NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AUTOMATING A STUDY QUESTION METHODOLOGY TO ENHANCE ANALYSIS IN HIGH LEVEL ARCHITECTURE

by

Michael W. Rauhut

June 1999

Thesis Advisor:	Arnold H. Buss
Second Reader:	William S. Murphy Jr.

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1999	Master's Thesis

4. TITLE AND SUBTITLE
AUTOMATING A STUDY QUESTION METHODOLOGY TO ENHANCE ANALYSIS IN HIGH LEVEL ARCHITECTURE

5. FUNDING NUMBERS

6. AUTHOR(S)
Rauhut, Michael W.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Training and Doctrine Command (TRADOC) Analysis Center Research Activities, Monterey
PO Box 8692
Monterey, California 93943-0692

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

The Department of Defense (DoD) uses simulation for many purposes. Early computer based distributed simulation support environments allowed individual models to communicate with each other but fell short of providing a general distributed simulation solution until the advent of High Level Architecture (HLA). HLA allows users to combine sub-models into one simulation, but it employs a subscription based communications scheme that did not exist in previous support environments.

Analysts often use a decompositional approach to identify measures of effectiveness (MOE), measures of performance (MOP), and data requirements for studies and tests. Fundamental study questions or operational requirements are decomposed until supporting data from tests and simulations are identified. This thesis formalizes this decompositional process, calling it the Study Question Methodology (SQM) and procedurally describes the steps all analysts should use to establish a clear audit trail from question to data inputs. It applies the SQM process to a study question relating to attack helicopters to demonstrate the dendritic (tree-like decomposition) approach. This thesis also provides a general solution for automating the SQM (ASQM) for use in distributed simulations that use the HLA. The ASQM enhances the analyst's pre-, during-, and post-exercise analysis. It provides the ability to answer study questions, establishes a clear audit trail, and helps fill an analysis tool void that presently exists in HLA.

14. SUBJECT TERMS
Distributed Simulation, Study Question, EEA, MOE, MOP, Modeling and Simulation, HLA

15. NUMBER OF PAGES
134

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239.18

Approved for public release; distribution is unlimited.

AUTOMATING A STUDY QUESTION METHODOLOGY TO ENHANCE ANALYSIS IN HIGH LEVEL ARCHITECTURE

Michael W. Rauhut
Captain, United States Army
B.S., United States Military Academy, 1989

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 1999

Author: Michael W. Rauhut

Michael W. Rauhut

Approved by: Arnold H. Buss

Arnold H. Buss, Thesis Advisor

William S. Murphy Jr.

William S. Murphy Jr., Second Reader

Richard E. Rosenthal

Richard E. Rosenthal, Chairman
Department of Operations Research

ABSTRACT

The Department of Defense (DoD) uses simulation for many purposes. Early computer based distributed simulation support environments allowed individual models to communicate with each other but fell short of providing a general distributed simulation solution until the advent of High Level Architecture (HLA). HLA allows users to combine sub-models into one simulation, but it employs a subscription based communications scheme that did not exist in previous support environments.

Analysts often use a decompositional approach to identify measures of effectiveness (MOE), measures of performance (MOP), and data requirements for studies and tests. Fundamental study questions or operational requirements are decomposed until supporting data from tests and simulations are identified. This thesis formalizes this decompositional process, calling it the Study Question Methodology (SQM) and procedurally describes the steps all analysts should use to establish a clear audit trail from question to data inputs. It applies the SQM process to a study question relating to attack helicopters to demonstrate the dendritic (tree-like decomposition) approach. This thesis also provides a general solution for automating the SQM (ASQM) for use in distributed simulations that use the HLA. The ASQM enhances the analyst's pre-, during-, and post-exercise analysis. It provides the ability to answer study questions, establishes a clear audit trail, and helps fill an analysis tool void that presently exists in HLA.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

As the Department of Defense (DoD) continues to look for ways in which to sustain its readiness and minimize costs, it faces the daunting task of training its forces for current operations and anticipating future operational requirements. With the advent of the modern computer, the DoD recognized that simulations provided a cost effective means to help train the force and conduct analysis.

Early simulations and computer models consisted of self-contained processes that were stand-alone systems. These early models were independent and did not communicate with each other. More recently, software architectures have arisen that facilitate distributed simulations using computer-networking technology. These allow a simulation system to be composed of numerous components or sub-systems that interact with each other over a computer network. The military's Simulation Networking (SIMNET) project demonstrated that distributed models could effectively interact. However, SIMNET had a very narrow focus that did not meet the needs of the various potential user groups. It provided the basic knowledge required to support future technological advances in model interoperability.

Within the military distributed simulation community, protocol-based distributed simulation support environments emerged. These environments generalized SIMNET technology, met the military's need for capturing interactions between all types of combat entities, and proved useful for analysis, training, mission rehearsal, and experimentation at the high-resolution level. However, these distributed simulation

support environments still fell short of a general solution to the problem of interoperability of distributed simulations.

The general solution for a distributed simulation support environment using a universal standard emerged with the High Level Architecture (HLA). HLA seeks to improve the interoperability of simulations in a distributed network and thus promote component reusability within the DoD. The components that are allowed to interact over the distributed architecture in HLA are called federates. Users combine federate components into a federation. The interacting federates are the combined sub-systems that establish a federation under HLA designed to model a larger system. HLA reduces the cumbersome overhead and interoperability issues of its predecessors by requiring users to subscribe and publish only those data objects each federate specifies before simulation execution. Even with the improved interoperability, however, HLA suffers from a lack of built-in analytical tools. Additionally, analysis in HLA remains difficult because of its unique front-end subscription requirements. HLA forces users into more disciplined, pre-execution analytical work than that of the former support environments.

Analysts should conduct a thorough, front-end analysis before conducting any simulation or experiment in support of a study plan. In practice, analysts sometimes look at computer files that capture data (data loggers) after the simulation and try to do some analysis. Under the former distributed simulation support environments this was possible. However, HLA federations do not maintain logger files because only data to which federates subscribe are available for analysis during and after run-time. Because

of the need for thorough front-end analysis in HLA, analysts need a methodology by which to decompose study questions into data level entities.

Analysts often use a decompositional approach to identify measures of effectiveness (MOE), measures of performance (MOP), and data requirements for studies and tests. In studies like Analysis of Alternatives (AoA) fundamental study questions or essential elements of analysis (EEA) are decomposed to identify data requirements. In Operational and Developmental tests, operational requirements are decomposed to identify test and simulation data requirements.

This thesis formalizes the dendritic process (tree like decomposition), calling it the Study Question Methodology (SQM) and applies the SQM process to a study question relating to attack helicopters in urban environments to demonstrate the process. This thesis also proposes a way to automate the SQM (ASQM) for use in HLA distributed simulations. The ASQM provides a general solution that automates the SQM for use by analysts in any HLA federation that uses any Federation Object Model (FOM). The thesis describes functionality requirements through the use of graphical user interfaces (GUI) for an ASQM that software developers can use to develop a user-friendly analysis tool for HLA distributed simulations. The ASQM allows the user to subscribe to needed data in HLA distributed simulations, presents real-time, viewable synthesis of the data, and augments post-processing capabilities that do not currently exist in HLA. Subsequent thesis and programming work will allow the analyst to document the decision process associated with the study question tree and its algorithms.

ACKNOWLEDGMENT

The author thanks Professor Arnold H. Buss, Professor Samuel Parry, and Major William S. Murphy, Jr. for sharing their wisdom and expertise and demonstrating patience during the writing of this thesis. The author thanks the personnel at TRADOC Analysis Center, Monterey, California and Fort Leavenworth, Kansas for their technical support and invaluable insights into the study question process. The author especially thanks his wife, Sandy and daughter, Hannah for their unconditional love, support and patience.

I. INTRODUCTION

A. BACKGROUND

Analysts use a decompositional approach to identify measures of effectiveness (MOE), measures of performance (MOP), and data requirements for studies and tests. In studies like Analysis of Alternatives (AoA) fundamental study questions or essential elements of analysis (EEA) are decomposed until supporting data from simulations are identified. In Operational and Developmental tests, operational requirements are decomposed until supporting data from tests and simulations are identified. This thesis formalizes this process, calling it the Study Question Methodology (SQM) and applies the SQM process to a study question relating to attack helicopters to demonstrate the dendritic approach. This thesis also describes how the SQM can be automated. The following sections provide important background information on High Level Architecture, the impact of architectures on analysis, the Analysis Federate, and the SQM.

1. High Level Architecture (HLA)

HLA is the current, Department of Defense (DoD) mandated architecture for use in distributed simulations (for a discussion/history of distributed simulations see APPENDIX B). The Defense Modeling and Simulation Office (DMSO) developed HLA in an attempt to establish a general solution to the problem of interoperability and reuse of different types of simulations within the DoD. [1] HLA consists of three components. The HLA rules govern the construction of participating simulations into one larger system model. The HLA Interface Specification governs how simulations interact with

an object model template. The HLA Object Model Template (OMT) provides a method for documenting key information about simulations and the system they are designed to replicate. [2]

The DoD goal was to provide architecture to support interoperability between various simulations across all domains in the DoD. [3] HLA provided the mechanism for interoperability of simulations in a distributed network. It promoted component reusability within the DoD by providing the common rule set absent from previous distributed simulation support environments (see discussion APPENDIX B).

The components that are allowed to interact over the distributed architecture in HLA are called federates. Simulations or federates are treated in the same manner as audio enthusiasts use stereo components. Stereo users who use the component approach have the ability to select particular components based on their preferences for high quality sound processing, price, functionality, and other factors. They are able to select the most appropriate components that will meet their individual needs. Likewise, simulation users who use the component approach can select the most appropriate simulation components and models to meet the required training or analysis need. HLA distributed simulation users like the DoD select federates or components based on user preferences for model fidelity, price, functionality, and other factors. They select the most appropriate federate that will meet their particular training or analysis needs. HLA users combine federate components together as interacting sub-systems of the larger system (federation) that is being simulated. This component based approach associated

with distributed simulations illustrates that the most appropriate components can be brought together to satisfy a particular need (see discussion APPENDIX B).

HLA improves model interoperability by establishing a common object reference system for all federates in the federation called the Federation Object Model (FOM). The FOM is a common domain dependent object model. Each federation has its own unique FOM with which all federates comply.

The common, domain specific FOM describes the accepted, shared data and naming convention used by the federates during federation execution. The Object Model Template (OMT) is the universal formatting standard that is used to document object class structure, data types, parameters, and attributes for the FOMs in the different user groups. Each federate uses a simulation object model (SOM) to document its capabilities. Each federates' SOM is an abstraction of the FOM and is documented in the OMT format. [4]

The Runtime Infrastructure (RTI) is the software that implements the HLA specification. The RTI serves as the equivalent of an operating system for the federation. It enforces the FOM's object model while providing interoperability services to the federates. Each federate must be designed to communicate with the RTI in accordance with HLA specifications and standards.

HLA uses a subscription based communications architecture. This differs from many of the other existing distributed simulation support environments that used a broadcast communications strategy (see DIS/ALSP APPENDIX B.3). By design, this subscription based communications structure reduces processing and bandwidth

requirements. The RTI ensures that federates receive only the information to which they have subscribed. Federates are only required to publish the information for which other federates have expressed an interest. This subscription based communications scheme significantly affects the manner in which data collection occurs for analysis purposes during a distributed simulation session. It precludes the use of network data logger files and associated post processing data analysis techniques other support environments use. Instead, HLA distributed simulation users develop alternative data collection and analysis strategies to support analysis in the HLA. These strategies must include provisions for the users to identify their data needs before the execution of any HLA distributed simulation session. Any data that is not subscribed to cannot be collected for subsequent analysis. Analysis in HLA thus requires significant front-end analytical effort because of HLA's unique subscription based approach.

2. Impact of Architecture On Analysis

Typically, distributed simulation support environments do not incorporate analysis tools, and HLA is no exception. The choice of the architecture used to support a distributed simulation session significantly influences the manner in which data collection occurs. Analysis techniques are similar across all architecture types and are merely repackaged into new tools to accommodate the different data collection techniques. After the data are collected, the analysis techniques are similar and architecture independent.

The Training and Doctrine Command Analysis Center (TRAC), Monterey is conducting research to develop data collection and analysis techniques tailored to support

the HLA. This research is essential because of the HLA provision that allows the use of domain dependent object models in the FOMs. This provision limits the potential usability of analysis tools developed to support an object model of a specific domain. TRAC, Monterey is developing the conceptual framework required to support the composability of data collection analysis tools across simulation sessions that use different FOM abstractions. This research supports the reusability of analysis tools and eliminates the requirement for each simulation community to develop specialized data collection capabilities.

The major difference between previous distributed simulation support environments and the HLA rests in the treatment of data. Previous support environments like Distributed Interactive Simulations (DIS) broadcast and received information in protocol data units (PDU) over wide or local area computer networks (WAN, LAN, respectively) at specified times or execution of an important state change event. PDUs facilitated data transfer between simulations that used different software by mandating the format of its PDU through its protocol standards. [5] HLA did not broadcast data as DIS did and thus placed additional subscription requirements on users.

3. The Analysis Federate

Since HLA's inception, TRAC, Monterey identified that HLA made it harder to develop a reusable tool to support data collection to perform analysis. Unlike HLA, DIS mandated all simulations use the same PDUs or call objects by the same name. This difference complicated the collection and analysis process in HLA. The subscription based approach of HLA made it harder to collect data. [6]

Since each FOM has unique objects, attributes, or interactions, one cannot simply map one FOM to another. For example, a public object model name and the associated attributes for a tank object in one federation can be quite different from those of another federation. The mainstream approach to address this issue has been to develop specialized data collection and analysis tools for each federation. Analysis in HLA required the development of separate, specialized analysis tools for each federation that used a different FOM abstraction. This approach complied with the HLA precepts and design principles. [7] However, it also frustrated general programming and analytical efforts because it violated the general precepts of reuse and limiting overhead.

TRAC, Monterey approached the issue differently. It discarded the HLA concept that called for a federate as an abstraction of a specific FOM. Instead, TRAC, Monterey developed a tool, referred to as the Analysis Federate that dynamically read the FOM and treated its contents as data. This allowed the Analysis Federate to aggregate data from multiple distributed simulations and provided the ability to perform real-time or post-exercise data analysis. This approach facilitated the development of an application that automated subscription, publication, data collection, and archival services. It allowed the tool to be used as a composable component in any federation that used any FOM. The tool provided users with a graphical user interface (GUI) that allowed them to automatically subscribe to and publish data and to implement the other HLA communications services without writing any computer code. The Analysis Federate thus insulated the user from the overhead associated with developing HLA federates. [8]

4. The Study Question Methodology (SQM)

The SQM is a general methodological approach to conducting thorough, front-end analysis. Its application facilitates problem definition, formulates important questions and issues, and helps identify data requirements before data generation. The SQM establishes the relationships between the question to be answered and the data required to derive the answer. Additionally, the SQM addresses the techniques and algorithms needed to synthesize the data into an understandable solution set. It is manually applicable as a front-end analysis tool in any architecture and context.

Analysts should conduct a thorough, front-end analysis before conducting any simulation or experiment in support of a study plan. [9] The success of the study largely depends on how well analysts define the problem and formulate important questions prior to collecting any data. [10] This drives what and how information is collected during the simulation or exercise. With existing distributed simulation support environments such as DIS, users could compensate for less than thorough front-end analysis by accessing data logger files after the run for any data requirements their pre-execution analysis failed to capture. Under HLA, users do not have this flexibility because of HLA's pre-execution subscription requirements. HLA forces users into more disciplined, pre-execution analytical work than that required by former support environments. The DoD need for formal, front-end and real-time automated analysis tools in HLA is significant.

B. PROBLEM

1. SQM Formalization

As currently practiced, the analyst's decomposition process is not formalized. Although decompositional methodologies are not new and are used in many areas of analysis, the analyst often applies the methodology intuitively. Done intuitively, the analyst may not capture the complexity of the posed study question to a sufficient degree that would allow later auditing. Ideally, analysts apply some version of the dendritic approach (tree like decomposition) before conducting training or exercises regardless of the architecture used. Analysts often look at the data logger files after the exercise and try to do some analysis. Analysts need a formal, decompositional methodology to allow them to quantify study questions and to establish an audit trail of their decomposition. That formal method, the SQM, is developed in this thesis.

2. SQM Automation

Analysts need an automated tool to enhance front-end, pre-, during-, and post-exercise analysis. Existing HLA analysis tools are currently limited to manual, front-end use. Analysts can use these tools to identify data requirements before HLA exercises, but they cannot provide real-time synthesis of the data during run-time. An automated version of the SQM would greatly enhance analysis in HLA distributed simulations. A prototype design for such a system, called the ASQM, is presented in this thesis.

C. STATEMENT OF THESIS

1. SQM Formalization

This thesis first discusses the formal development of the SQM. It presents the general methodology analysts should use when conducting front-end analysis in any analytical situation to include distributed simulations and field exercises.

The thesis uses a study question that involves an AH-64D attack helicopter operating in an urban environment to illustrate the application of the steps in the manual SQM process. Current industrial and military interest in the posed study question led to its selection as the SQM application example. The study question is decomposed to a level sufficient to facilitate this illustration.

2. SQM Automation

The thesis also provides a systems design for automating the SQM, the ASQM, that enables analysts to establish a clear, automated audit trail from posed study question to data requirements. It accomplishes the task of establishing relationships between study questions and the data required to derive the answer. It provides the user the flexibility to identify the algorithms needed to establish a solution set. Used in conjunction with the Analysis Federate, the ASQM facilitates analysis in HLA distributed simulations by providing the user with a thorough, front-end analytical tool. Additionally, it allows the user to subscribe to needed data in HLA distributed simulations, presents real-time, viewable synthesis of the data, and augments post-processing capabilities that do not currently exist in HLA.

This systems design can be used to develop a requirements document for use in developing a user-friendly ASQM analysis tool for HLA distributed simulations. The automated solution integrates the ASQM and Analysis Federate. It capitalizes on the FOM reading capabilities introduced in the Analysis Federate research and point and click mapping between the FOM and the SQM tree. TRAC, Monterey is currently preparing a requirements document and attempting to obtain funding needed to implement the results of this thesis.

The following chapters discuss the salient formalization and automation issues identified in this chapter. Chapter II describes the general dendritic approach of the SQM. It describes the SQM tree and key components of the SQM. Chapter III applies the material of Chapter II to a proposed study question relating to the use of attack helicopters in anticipated urban environments in the 21st Century. It also describes the relevance of urban environments and provides background information on the AH-64D. It concludes with some general insights of the SQM process itself and considerations for attack helicopter employment in urban environments gathered during the application. Chapter IV addresses the automation implications of the SQM and proposes a general automated solution (ASQM). It discusses design precepts, proposes a method by which to populate the SQM tree with HLA distributed simulation data, and demonstrates functionality requirements of the ASQM using graphical user interfaces (GUI). The body of this thesis concludes with Chapter V that summarizes the information gleaned from this study. Attached appendices provide additional information and are referenced, when appropriate, throughout the body of this thesis.

II. STUDY QUESTION METHODOLOGY

This chapter describes the formalization of a process analysts can (and should) use when investigating a posed study question. It describes a decompositional approach, referred to as the SQM, that allows analysts to establish a clear audit trail from posed study question to data requirements. Section A introduces the SQM tree and the overall SQM process. Section B discusses the study question, the importance of context, the iterative, often repetitive decompositional steps, and finally arrives at the data required to answer the questions posed during the decomposition. This chapter concludes with Section C that identifies some important implications of the SQM and ASQM.

A. DESCRIPTION

1. Background

The Army trains its forces, procures modern equipment, and experiments with its force structure to ensure it can "preserve the peace and security, and provide for the defense of the United States." [11] The U.S. Army executes training and conducts experiments for the explicit purpose of improving its readiness and ability to perform its current and future missions. Examples of specific objectives include identifying a force structure for future Army requirements, assessing a unit's combat readiness, or deciding whether to procure a new weapons platform.

In the quest to answer study questions, the analyst articulates objectives, establishes evaluative criteria or measures, identifies the means of data collection, and analyzes the data once it is collected. The decision-maker uses the analysis to support the

decision-making process, assess whether the stated purpose was achieved, and identify the conditions under which the question may be answered differently.

Experiments and training provide the military with feedback that it uses to assess its readiness and operational requirements. Analysts use varying levels of abstraction to assess training and experimental results ranging from "set-piece" standards to ad-hoc study questions. Soldiers serving as unit observers must access or critique performance and conduct after action reviews. These observers/controllers use time-tested, structured techniques to arrive at an assessment of the stated study objective. For them, the analysis is clear and very definable. For analysts investigating broader issues, there is a need for a more abstract method of analysis.

2. SQM Roadwork

The trained analyst needs a formal methodology to address study questions. The SQM supports this need by formalizing the intuitive process typically followed by analysts during a study. It procedurally describes the steps all analysts should use to establish a clear audit trail from a study question to its data inputs. The Army decomposes this process by identifying Essential Elements of Analysis (EEA), Measures of Effectiveness (MOE), and Measures of Performance (MOP). With only intuitive processes, analysts may lack the structure required to audit the process, identify potential problem areas, and ensure the study is executed successfully.

3. The SQM Tree

The SQM systematically dissects the study question into identifiable data collection requirements (Figure 1). [6] The following description of the SQM uses the Army's decompositional terminology that includes EEAs, MOEs, and MOPs.

Analysts using this process first identify the decision-maker's questions or issues. Next, analysts select study objectives that address those issues. They then identify EEAs or questions that are more specific that require answers in order to meet those study objectives. Next, analysts identify the MOPs and MOEs needed to answer the EEAs. They identify, collect, and use required tools and data to develop the measures. Finally they use the results to answer the specific questions and decision maker's issues. [9]

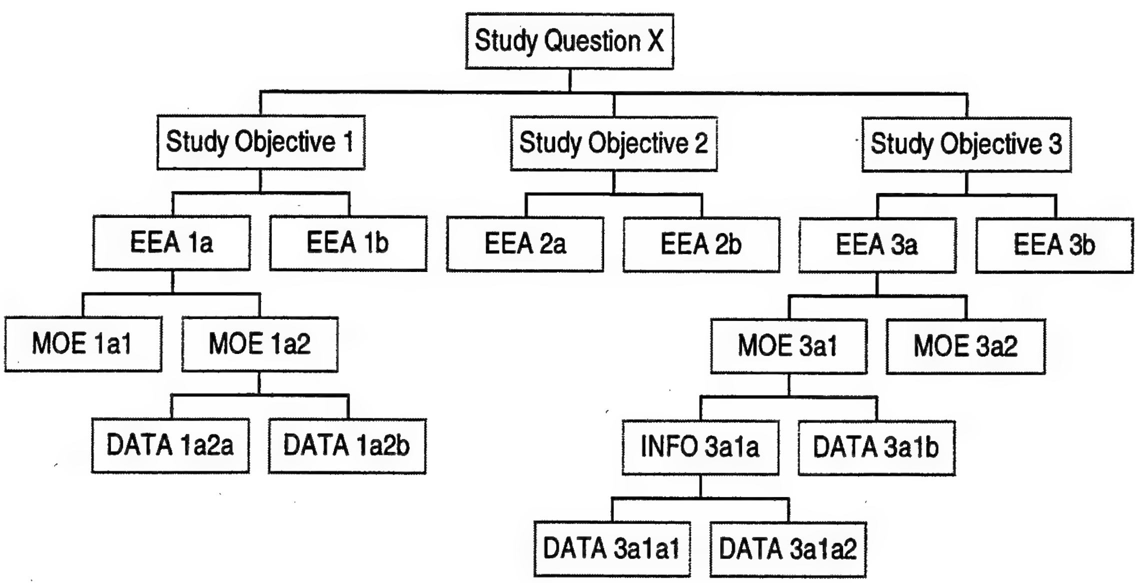


Figure 1. Study Question Methodology Tree

Through this decomposition process, the SQM creates logical and mathematical relationships between parent nodes and their children. The methodology is thus a

dendritic analysis technique similar in appearance to a rooted tree. The rooted tree, or aborescence, has a distinguished node called the root. In the SQM, the overall study question serves as the root node. The analyst develops subsequent questions from the root node to further define or quantify the study question. The Army abstraction calls this first branching creating sub-issues or study objectives. These first "children," in turn, serve as parent nodes for further questions that add greater measurability to the parents. The analyst continues this decompositional approach and may identify critical nodes that must be answered to satisfactorily answer the overall study question. The Army abstraction calls these critical nodes EEAs. The children of these critical nodes are typically measures of the study. Again, the Army abstraction calls these children MOEs or MOPs. The decomposition continues until the tree terminates with data (leaf node) level children, the parent of whom is typically a MOE or MOP. The depth of the SQM, defined as the number of edges on the path to the root, can vary with the number of children present in the tree. Even within the tree, branches can have different depths based on the types of questions posed and the measurability of the question. The height of the SQM tree, defined as the largest depth of any node in the tree or the maximum of all depths, also varies with the number of children. Each of these branches and parent-child nodes relates to each other and helps decompose the study question into quantifiable entities.

4. Benefits of the SQM

Analysts provide a quantitative basis for decision-making. The analyst best serves the decision-maker by developing and using quantitative tools. These tools (one of which is the SQM) and the resulting analysis provide the decision-maker with a more rigorous, analytically based reasoning process than that available from experience alone. What at first may appear intuitive to the decision-maker may in fact be counter-intuitive after quantitative analysis.

The SQM serves as a tool by which analysts can show decision-makers, through a clear audit trail, how the analyst answered the question. It helps decision-makers recognize and account for the risks under which they make decisions. Further, the SQM promotes a better appreciation for the complexity of the issue under investigation. It requires analysts to decompose the study question into digestible pieces for the decision-maker before the simulation or experiment is conducted. The SQM demands analysts "functionally decompose" from the abstract study question level to the detailed data or information level. [10] Application of the SQM can help determine how "answerable" a particular question is and what tools can best provide the data required to make the decision.

B. SQM COMPONENTS

A thorough decomposition that identifies and subsequently captures the required data will provide an answer to the study question. The synthesis process introduced in Section II.A.3 makes it possible to answer the study question because the SQM tree's children all become data once the leaf nodes of the tree are populated. During synthesis

up the tree, each node uses its children's data in its algorithm to convert the node into a piece of data itself. As data goes up the tree, each node becomes data in a different form. For example, a node with number inputs like "0" or "20.0" may translate those numbers into a text-based representation such as "True" or "Excellent." This translation now makes the node a piece of text-based data. Ultimately, all children on the SQM tree become data with which to answer the study question (root node).

The process described in the previous paragraph requires the analyst to identify a question, algorithm, and parameters for each node in the SQM tree. The analyst must capture these important characteristics for each node from the study question down to and including the data (leaf) nodes. The three important characteristics of each SQM tree node are depicted in Figure 2 below.

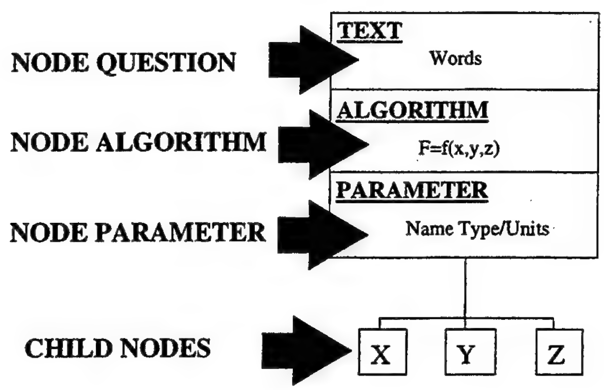


Figure 2. Node Characteristics

1. Node Algorithms and Node Parameters

This section provides a general discussion of node algorithm and parameter development. In practice, the analyst can not develop the node algorithm and node parameters without a node question.

The node question provides the user and decision-maker with a clear, text-based view of the decomposition. The node algorithm or equation combines the children in such a way that the node produces an answer to the question of that node.

The analyst can use the SQM to decompose the study question to a level from which data becomes collectable. This decomposition process is analogous to the analyst "climbing down the tree" from the root node until reaching collectable data. Part of this process is establishing an algorithm for each node that produces an answer to that node's question. The algorithm combines the inputs from the children nodes to produce the node's answer. The following section describes the text-based node question representation of the SQM.

2. Node Questions

The node question provides the user or decision-maker with a clear, text-based representation of the decomposition. Recall that each SQM node question is also going to have an associated algorithm and parameter. This section describes the SQM process used to develop the hierarchy of node questions independently from the node algorithms and parameters.

a. The Root Node

Study questions typically lack precision or quantification and are not directly measurable. Examples of study questions include: "What should the Army's Strike Force look like?"; "How many ships does the Navy need in order to meet operational requirements?"; "Does the Joint-Strike Fighter fulfill its operational characteristics requirements?"; and "Does the Marine Corps possess sufficient close air

support assets?" The first step for the analyst is to identify and define the study question in its proper context.

There is a context associated with any posed study question. This context serves as the lens through which the analyst views all subsequent decomposition and analytical steps and provides the required structure to successfully conduct the study. Understanding the context of a particular study question is important because the analyst uses it to develop a common understanding of the problem.

Context within the DoD often involves readiness and the ability to perform wartime missions (see discussion APPENDIX C). Each service articulates the way it envisions accomplishing its mission through its published doctrine. As each service describes and outlines its overall objective, they establish requirements. Each subordinate level of command within the services must meet these requirements in order to accomplish their missions. These top-down, concept based requirements focus on military doctrine and provide context to all subordinate levels of command.

Measurability increases as a study question is decomposed or if a study question is posed at a lower level of abstraction. For example, a study question that asks, "What should the Army Strike Force look like?" must identify the roles and missions of the Strike Force as well as address how to access the effectiveness of the Strike Force during analysis. At a lower level of abstraction, a study question may ask, "How much weight can the average light infantryman carry in a rucksack?" This study question can be decomposed more easily because it is posed at a level of abstraction much nearer to collectable data than that of the Strike Force question.

b. Step One: The Study Question Context

Using the SQM, an analyst first identifies the level of abstraction at which the study question is posed and define the study question within that context. He then looks for vague terms in the question that are not directly measurable and require further definition. The definition of these terms must address the context of the question. As the decision-maker poses the study question, the analyst goes through a thought process that includes questions like: "What does a particular term mean?"; "Is the term measurable?"; "How will I decompose the question into supporting issues?"; and "What kinds of issues tell me something about the question?"

c. Step Two: Quantitative Decomposition

The next step in the SQM is the further decomposition of the study question into sub-questions by trying to determine the components of the question at hand and the functions required to achieve the desired goal. In DoD applications, analysts may try to define the imprecise terms used in a particular study question by applying doctrinal terms and definitions. Doctrinal manuals at the strategic, operational, and tactical levels of war help the analyst understand the functions required for mission accomplishment. Analysts may find that initial sub-issues or subsequent questions remain subjective in nature and can not be directly measured from available data; however, analysts can identify indirect measures to help them answer these subjective questions. The indirect measures then serve as the children nodes on the SQM tree. These subsequent questions serve as a perceptual or conceptual guide to further analysis.

They focus the information requirements needed to later synthesize data back up the tree and recommend to the decision-maker the answer to the overall question. [10]

The analyst must continue the iterative decompositional process and identify nodes of particular relevance to the study. These critical nodes are important because without them, the study question can not be answered to the satisfaction of the decision-maker. For example, the Strike Force study question proposed earlier may decompose into firepower, command and control (C2), and logistics issues. The decision- maker may be willing to forgo exhaustive analysis of the logistical issues in order to focus on the composition of the fighting force and how to control that force. If so, the analyst would consider the firepower and C2 nodes as critical nodes.

The analyst continues the decomposition of critical nodes into more quantifiable measures. Analysts must identify measures (MOEs or MOPs) that individually or collectively answer the parent critical node (EEA). This may require still more decomposition or will fall out from the context of the critical node question. Typically, a measure node (MOE/MOP) serves as the parent for subsequent data (leaf) nodes. Step Two is complete when the analyst identifies that the node question he has posed can be answered with available data.

d. Step Three: Identifying Data (Leaf Node) Level Requirements

The analyst eventually arrives at entity-level children nodes where data is available for collection during the actual experiment. These data level nodes are the analysts' most objective measure within the whole decomposition process. Occasionally, analysts who developed what they considered the "perfect" measure (MOE/MOP) may

find the required data inputs are not available for collection. This type of measure serves no purpose. Analysts who find the data unavailable are forced to abandon the measure and develop alternative measures to answer the question from a different perspective. The analyst must return to the data (leaf) node's parent measure to try another decomposition until he or she finds suitable data to collect. It may be necessary to develop entirely different parent and grandparent questions in order to address the decision-maker's study question. [9]

e. Step Four: Synthesis

Once this decomposition is complete, the analyst must synthesize from the deepest child back to the root node. The purpose of this synthesis is to answer the question by confirming the collection plan will be capable of providing all of the data needed to answer the study question. Synthesis starts with the leaf node data and collapses the tree upward using that node's algorithm until the study question is answered. Synthesis also helps confirm that the data level requirements are collectable and do in fact answer all of the nodes in the SQM tree in the manner the analyst expected. This synthesis audits the process and helps the analyst identify shortcomings. An explicit example of SQM synthesis is shown in Section III.C.2.

The analyst must provide feedback to the decision-maker in cases where the decision-maker has specified objectives or issues that can not be answered with data that is generated and available for collection during the exercise. This allows the decision-maker to better understand the risks under which he or she is making the decision and provide further guidance. Ideally, the analyst can develop alternate

questions (EEAs, MOEs, MOPs, etc) to address discrepancies. In reality, the analyst may be forced to return to the decision-maker and explain that neither a particular question can be quantified, nor a suitable alternative found. If the decision-maker is aware of the problem, he or she can accept the analyst's work and ignore that portion of the tree. The decision-maker can accept the work with qualification and continue the study including the questionable portion of the tree, understanding the suspect nature of the tree. Alternatively, the decision-maker can reject the analyst's work and start over. [9]

C. SQM AND ASQM IMPLICATIONS

1. SQM Applicability

The SQM supports both the analyst and decision-maker by formalizing the intuitive process currently used by some analysts and by decomposing study questions into data nodes. It establishes a clear audit trail, ensures the analyst considers whether a particular model is capable of answering proposed questions, conserves valuable time and resources, and does not create false expectations on the part of the decision-maker. It allows the analyst to determine if the study will accomplish stated objectives before the exercise. This minimizes situations where the analyst and decision-maker are surprised to learn after the fact that the study question can not be answered because the data was not available or not collected.

The SQM is applicable any time a posed study question is far removed from the data required to answer that question. It provides a methodology to figure out how to get to data. Its potential uses are wide-ranging and not limited to the Army applications

described in this chapter. Analysts working for any organization within any context could apply the decompositional approach of the SQM to a variety of problems.

2. ASQM Applicability

There are numerous benefits to automating the SQM for application in HLA. The greatest benefit to automating the SQM in HLA is the real-time visual display of the analyst's front-end analysis efforts. The ASQM automatically connects the data to the analysis so that the decision-maker and analyst can watch in real-time the SQM populate and produce an answer to the study question.

Analysts need more than a partially automated front-end analysis tool that allows them to apply the SQM in a user-friendly interface. Such a tool would only meet an information management and data requirement identification need. Analysts need a single tool that helps them identify what to collect, write required algorithms to provide answers, and select the data with which to use in those algorithms. Application of a methodology like the ASQM is one way an analyst can ensure that the HLA subscription requirements will be comprehensive.

3. ASQM Functionality

Analysts, particularly Army or DoD analysts, need to know what happens over time and are particularly interested in changes. Decision-makers like to understand causes of changes in state of the entities with which they operate. This provides them a "temporal context." The temporal context is important because it provides more insight that a static one. Since Army questions are often time dependent, the ASQM tool must capture changes that occur over time. The ASQM needs a scheme to capture change

information rather than just capture snapshots of situations in time. This requirement demands users be able to state thresholds to capture required information.

The SQM and ASQM provide the analyst with a methodology capable of decomposing a study question to its elemental data level. This chapter described the steps of the SQM and the benefits of using such a methodology to determine data requirements. The next chapter will illustrate the application of the steps in the manual SQM process to a study question.

III. SQM APPLICATION TO URBAN SCENARIO

This chapter applies the SQM described in Chapter II to a study question relating to attack helicopters. Section A describes the relevance of anticipated urban operations in the 21st Century and explains the impact of the urban environment on military operations. Section B provides background information on the AH-64D attack helicopter. Section C illustrates the application of the SQM by posing a study question relating to the relevance of attack helicopters in anticipated urban environments in the 21st Century. This chapter concludes with Section D that identifies some important implications gleaned from the application of the SQM to the attack helicopter question.

A. URBAN WARFARE/MOOTW RELEVANCE

Increased operational tempo, strict budget constraints, and evolving doctrine, makes it imperative for the DoD, and in particular the U.S. Army, to investigate the combat roles and missions of its current inventory. These constraints require the Army to investigate hardware and operational requirements of future combat forces. [12] The U.S. Army and Marine Corps anticipate increased operational requirements in urban environments and doctrinally address missions ranging from all out urban combat to Military Operations Other Than War (MOOTW). They expect operational employment in urban environments to increase in the next millenium. Marine analysts predict seventy percent of the world's population will live in urban areas by the year 2020 due to increased migration to the cities. [13]

This migration to the cities brings with it political, cultural, religious, and social tensions that often lead to conflict. Population shifts place demands on scare resources that many nations' urban centers can not support. Quite naturally, people and organizations within these urban centers seek to accomplish competing objectives. Governments around the world are concerned about the proliferation of inexpensive weapons of mass destruction. The Army and Marine Corps understand the certainty of military operations in urban environments in the 21st Century. [13]

The Army and Marine Corps seek to address these issues by refining and modifying their doctrine and tactics. The independent and shared efforts between the Army and Marine Corps, including the Center for Army Lessons Learned and the Marines "Urban Warrior" experiments, demonstrate the seriousness with which these Services take the urban warfare issue. The Army and Marine Corps share an interest in ensuring their tactics, techniques, and procedures take advantage of advances in technology. Additionally, they strive to ensure that their doctrines and operational concepts remain relevant and address the realities of anticipated urban operations in the 21st Century. [14]

Urban warfare challenges the U.S. Army and Marine Corps differently than other types of operational environments. Because of its unique physical characteristics, the urban environment limits many of the capabilities military systems and forces seek to use to advantage in more open terrain. The physical characteristics of the urban environment collectively impede mobility, frustrate communications, and limit direct and indirect fire systems in ways that make urban combat unique to all other environments.

Many physical characteristics of urban landscape significantly affect military operations. Some of these physical characteristics and their impact on military operations are described in this paragraph. One characteristic is referred to as the "Urban Canyon." "Urban canyons" are created by the close proximity of buildings in generally linear patterns [15]. Another important characteristic is subterranean space that is somewhat unique to urban environments. Complex subterranean avenues of approach such as sewers, subways, cellars, and utility systems greatly impact urban operations. Other characteristics of urban environments include impeding structures, dust, and smoke. These tend to limit observation and fields of fire and cause shortened target exposure times. Combat is focused at the small-unit level and is characterized by isolated fighting because of the "compartmented" nature of the terrain. There is an increased need for munitions and special equipment caused by the shorter target exposure times and need for troop mobility assets, respectively. Radio and visual communications suffer degradation due to the obstructive nature of most buildings. Command and control proves difficult because of this degradation. [14]

Urban warfare also differs from other warfare environments in that it places greater restriction and control on the use of combat power. Urban operations typically involve engagements of enemy forces amidst or near a civilian, non-combatant population. Since the law of war rightly prohibits unnecessary injury to civilians and needless damage to property, rules of engagement (ROE) force the surgical application of combat power in urban environments. [14] These restrictions often prove necessary to support higher political and social objectives. Concerns over increased incidence of

fratricide also make urban environments more difficult for military operations. [14] The density and multi-dimensional nature of the urban landscape, restrictive mobility corridors, munitions effects, C2, and ROE issues all play havoc on what the Army historically held as its focus for employment prior to the fall of the Berlin Wall.

Under the former Cold War paradigm the U.S. military, particularly the Army, focused its attention on a large scale, ground war in Europe. It dedicated vast resources to the protection of Western Europe from an anticipated Soviet invasion. The Army's Airland Battle doctrine treated built-up areas as "no-go" or very restrictive terrain. It advocated bypass of urban areas if the offensive situation permitted. Airland Battle doctrine also advocated establishing the defense within urban areas only out of operational necessity. It rightly recognized that urban areas typically require large commitments of soldiers and other resources to conduct successful combat operations. The doctrine suggested avoidance of urban areas if operationally possible; however, it also identified the situations and conditions under which combat in urban areas was advantageous. [14]

The Army was slow to recognize the magnitude of the changing world environment. It continued to treat urban areas as "restrictive terrain" during the transition from Airland Battle Doctrine to operations in MOOTW scenarios. [14] Neither Airland Battle nor MOOTW advocated fighting in urban environments. However, they both addressed war fighting issues and the tactics, techniques, and procedures required to fight and win in urban terrain. This treatment of urban areas illustrates that the Army did not,

and perhaps could not, realize the range of missions and roles it would perform in support of operations other than war in urban environments.

MOOTW demands that Army and Marine forces rapidly deploy and accomplish a wide range of missions in and around urban terrain. Unlike the past, the Army currently faces a much more dynamic and less predictable world in which urban centers play a huge role. U.S. operations in Somalia, Haiti, Bosnia, and involvement in Kosovo demonstrate the need for the Army to continue to analyze the way it conducts operations and fights wars. Coupled with decreases in force structure, the requirement to investigate the operational capabilities of its current and future combat multipliers demands careful consideration.

There is a great need for multi-capability systems under this new paradigm. Organizations like the Army can not afford to use assets in limited roles unless evidence restricts the asset's use to such roles. It is in the Army's best interests to investigate all methods it uses to accomplish missions and to analyze the roles of its personnel, equipment, and organizations in all terrain, especially urban environments. Since urban environments prove to be the focus of future conflicts, it is worth investigating the performance of weapon systems in such environments. For illustrative purposes, this thesis addresses attack helicopter operations in anticipated urban environments in the 21st Century.

Under the former Warsaw Pact threat, the attack helicopter served as a tank or armored formation destroyer. It continues to be an asset that maneuver commanders, typically corps and division commanders, use to penetrate enemy formations in depth and

to strike highly valued targets. [16] Because of the nature of the urban environment and the traditional role of attack helicopters in Europe as a tank killer, Army doctrine did not fully explore the ways that attack helicopters could help ground forces operating in urban environments. The role of attack helicopters in urban areas was not clearly defined. The fall of the Soviet war machine, the rise of MOOTW missions, and the joint nature of warfare caused the Army to begin developing doctrine for other attack helicopter roles. [16]

B. AH-64D LONGBOW ARMY ATTACK HELICOPTER

The AH-64D Longbow attack helicopter (Figure 3), the Army's premier attack helicopter, was selected to illustrate the application of the SQM in this thesis. The AH-64D was selected because it faces certain employment in urban environments in the 21st Century. Its predecessor, AH-64 Apache served first in Panama and later in the Gulf War where 288 Apache helicopters demonstrated their combat capabilities in numerous, attack roles in a desert environment. [17] The AH-64D possesses greater, more advanced war-fighting capabilities and in different variants performs the Army's attack helicopter role around the world.

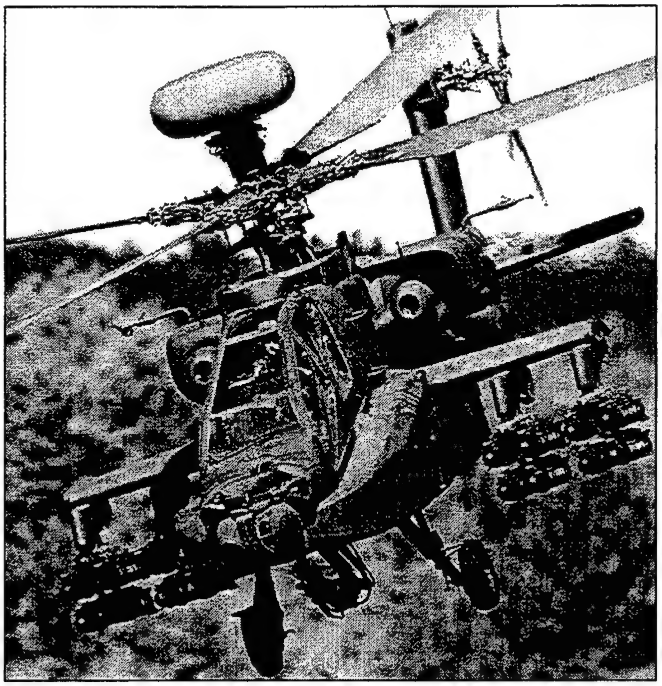


Figure 3. AH-64D Longbow

The AH-64D was developed and built by Boeing. Its advanced direct fire systems, target acquisition and night vision, electronic warfare (EW), and structural features make it a formidable weapon system. The AH-64D caters its weapons mix to its assigned combat mission and takes advantage of advanced targeting and acquisition systems to dominate the battlefield. The Target Acquisition Designation Sight (TADS-AN/ASQ-170) and Pilot Night Vision Sensor (PNVS-AN/AAQ-11) allow the AH-64D to search, detect, and recognize targets in numerous environments. The TADS provides direct view optics, television and three fields of view forward-looking infrared (FLIR), and includes a laser range-finder/designator. The pilot and copilot (gunner) control the PNVS using a monocular eyepiece as part of the Helmet and Display Sighting System (HADDS). The PNVS is located on the nose of the AH-64D above the TADS and consists of a FLIR. [17]

The Longbow's arsenal consists of a 30mm automatic chain gun under its nose which achieves a 625 rounds per minute rate of fire with a 1200 round carrying capacity. It can carry air-to-air missiles like the AIM-9 Sidewinder, Stinger, Minstral, and Sidearm to engage enemy aircraft. For area suppression missions, the AH-64D can use its 2.75-inch Hydra 70 rockets carried in pods from hard-points on its arms. Other direct fire weapons include air-to-surface AGM-114D Longbow Hellfire missiles with a range of 8 to 12 kilometers (km) that use the millimeter-wave Longbow fire control radar to assist in engaging targets. [17]

For passive identification and location of threats emitting radar, the fire control radar incorporates an integrated radar frequency interferometer capable of distinguishing targets from ground clutter and operating under low visibility conditions. It is less susceptible to electronic countermeasures because of its narrow beam-width and short wavelength. Processors in the Longbow scan, locate, and determine speed and direction for up to 256 targets.

The AH-64D also possesses advanced EW and survivability characteristics. Its EW capabilities include the AN/APR-39A (V) radar-warning receiver, AN/ALQ-144 infrared countermeasures set, AN/AVR-2 laser warning receiver, AN/ALQ-136 (V) radar jammer, and chaff dispensers. The AH-64D is survivable from rounds up to 12.7 mm in size. Kevlar seats protect the crew stations, and boron shielding protects the cockpits. The rotors are tolerant to hits by 23 mm rounds. The AH-64D is the most survivable and lethal attack helicopter in the U.S. Army and perhaps the world. [17]

C. SQM APPLIED

Interest in defining the roles and missions of attack helicopters continues both in the Army and the nation as these assets deploy in support of military operations to places like Kosovo. The Army investigates what types of helicopters can best accomplish specified roles and missions. It intends to field more AH-64D helicopters to perform the attack helicopter role into the 21st Century. The Army identified the need for another helicopter and is currently testing the Commanche. The Commanche's specified, short-term purpose is not to replace the AH-64D. Rather, it is designed to perform in a reconnaissance role.

The Army at some point in the future may have to ask if it can afford two highly advanced, expensive helicopter systems or if it must rely on one, multi-role aircraft. It could analyze each platform's effectiveness in performing attack and reconnaissance roles to determine if one or the other aircraft should become its one, multi-role aircraft. The larger study question might be which of the two helicopters to select. Analysis could be conducted with the AH-64D and Commanche using the same scenario, terrain, and other pertinent factors to compare performance. Since the Commanche is not yet fielded or widely familiar, this thesis develops a study question based on the future use of the AH-64D alone. It investigates relevance of the AH-64D to combat in the 21st Century, particularly anticipated operations involving urban terrain.

The decision-maker's perspective on attack helicopter operations, particularly those operations in urban environments, varies based on the decision-maker's background, position, and familiarity with helicopter employment. Army decision-

makers may ask what roles attack helicopters play in anticipated current and future operational scenarios. Defense industry companies such as Boeing share the Army's interest in investigating current and future roles for the AH-64D. For Boeing and its suppliers, the relevance of the AH-64D to urban warfare in the 21st Century influences the amount of revenue they earn through AH-64D sales. Although the perspectives of industry and military decision-makers are not orthogonal, they differ in their bottom line. Certainly, a mutual codependence recognizes the need for useful, effective weapons platforms like the AH-64D.

Both the U.S. Army and Boeing possess a stake in helicopter operations in the 21st Century. From the Army's perspective, any attack helicopter must be capable of operating effectively in urban environments in support of MOOTW and conventional urban warfare. From Boeing's perspective, it can recommend upgrades to existing aircraft or propose new aircraft that can meet the Army's mission requirements with the underlying interest of making a profit.

The Army may pose many legitimate study questions to examine the roles advanced attack helicopters can perform in urban terrain. For example, Army decision-makers may ask, "Is the AH-64D Longbow Apache relevant to anticipated urban operations in the 21st Century?" This serves as the study question this thesis uses to illustrate the application of the SQM. This study question is the root node of the SQM tree.

1. Study Question Illustration

Recall that at each step of the process and for each node in the SQM tree, the user must define a node question, a node algorithm, and the node parameter. The SQM tree nodes need equations to ensure the correct data is collected and synthesized. Because this portion of the thesis focuses on the hierarchy of questions, it does not illustrate the algorithms used at each node.

a. Step One: The Study Question Context

As discussed in Chapter II, the analyst using the SQM first identifies and seeks to understand the context of the question. In this particular example, the analyst develops the context by understanding the importance of the decision and the way in which the AH-64D helps organizations accomplish their missions. The AH-64D typically serves as an asset for higher levels of command (Corps, Division). Additionally, it is employable in urban areas in support of special or conventional, joint or contingency operations. The analyst may decide that definition or decomposition needs to address the doctrinal impact of the AH-64D on such missions or employment methods. Doctrine provides the analyst the context in which to begin decomposition. At the micro-level, the AH-64D influences the tactical level of war. Although a headquarters at the operational level (for instance, a JTF headquarters) may control its parent unit, it executes its missions at the tactical level of war. The analyst may conclude that decomposition must address the objective or mission of the AH-64D in its tactical level, doctrinal context.

Further, the analyst using the SQM may note the ambiguity of the term "relevance," since it lacks definition and quantification. As the decision-maker poses the question, the analyst thinks, "What does relevant mean? Is relevant measurable? How will I decompose the issue? What kinds of data tell me something about "relevance?" The analyst should anticipate the types of questions the decision-maker will have when presented answers.

The analyst should use current doctrine as the context in which to decompose the study question by identifying the functions in which AH-64Ds perform. He or she should identify sub-issues and associated questions from the root node. This first branching from the root node serves to add clarity to the study question.

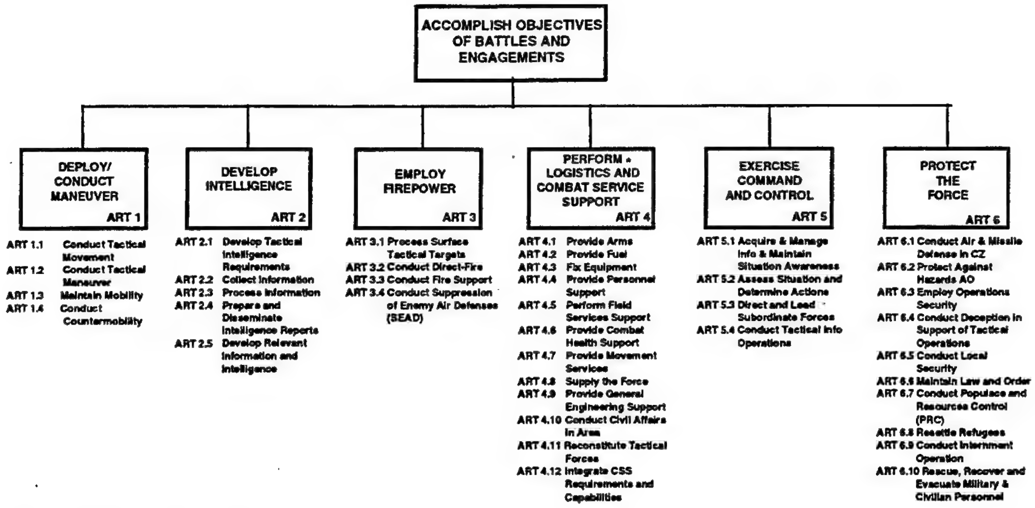
b. Step Two: Quantitative Decomposition

(1) Decomposition to Sub-Issues

While "relevance" in anticipated urban operations in the 21st Century is difficult to quantify, the analyst can begin by understanding its meaning from relevance on today's battlefield. The analyst can add to this base knowledge while going through the process to better articulate to the decision-maker how relevance was determined and to help ensure that the decision-maker's thoughts were addressed. There are numerous doctrinal manuals and references from which to define relevance for a combat system in warfare. One such document is the Army Universal Task List (AUTL) that lists strategic, operational, and tactical levels of war tasks in a comprehensive structure. It serves as a common reference system for Army, Joint, and multinational forces at all three levels of war. [18]

The AUTL articulates the Army's role in achieving objectives of battles and engagements at the tactical level, of subordinate campaigns/operations at the operational level, and of theater and national strategy at the strategic level of war. The AUTL describes the tasks Army soldiers, systems, units, and higher-echelon organizations perform. The analyst can use it as a reference to investigate doctrine, training and operations, organizations, leader development programs, and materiel. It provides the analyst a basis for establishing performance standards necessary for successful execution of Army missions or operations. Additionally, it provides a framework to support studies and analyses, scenario development, materiel system requirements, doctrine development, training and education, test and evaluation, readiness assessment, operations planning, strategy development, and operations templates. [18]

The analyst can use the AUTL to decompose the study question into study objectives and issues as part of the SQM process. The AUTL when used in this manner can help the analyst quantify the term "relevance." The analyst can use the AUTL to identify the roles and missions that the AH-64D performs to support Army units in urban environments and to accomplish current and future missions. The usefulness of the AUTL to support this process is illustrated by the tactical level tasks that were extracted from the AUTL document (Figure 4). [18]



* Also referred to as Tactical Logistics (see FM 100-5).

Figure 4. AUTL – Tactical Level

The analyst decomposes the study question into SQM sub-issues by branching from the root node (Figure 5). The analyst completes this step of the decomposition process by using the AUTL to assist in the identification of potential SQM sub-issues. Note that the root node is the study question from Section III.C and is not a component of the AUTL.

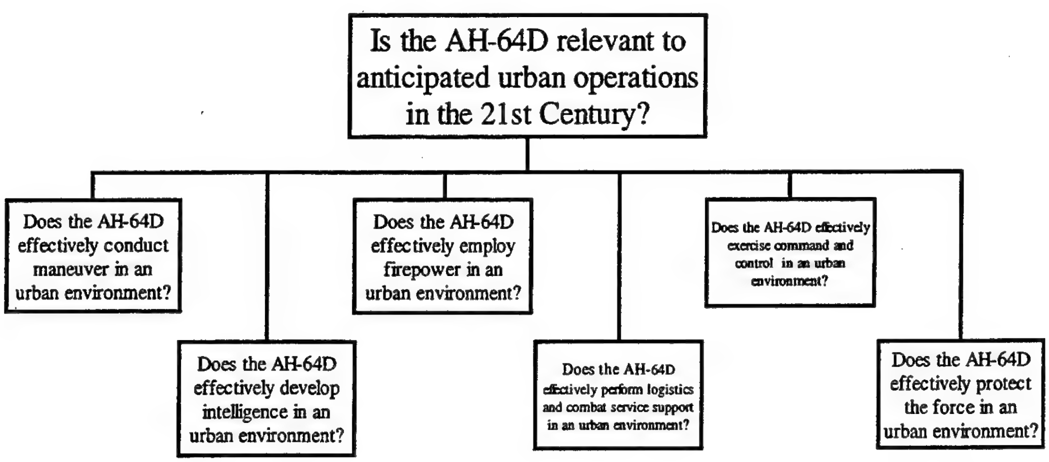


Figure 5. AH-64D Study Question Sub-Issues

The decomposition depicted in Figure 5 identifies operational task areas to focus collection and analysis efforts into different areas of interest. The initial children of the root node (sub-issues or objectives) serve as indirect measures of the parent (study question). They also serve as the perceptual or conceptual guide to further decomposition and focus the information requirements into more quantifiable areas of interest. However, they still lack direct measure, and thus the analyst must continue application of the SQM.

Figure 6 illustrates how the analyst has chosen to distinguish between today's technology and future technology to establish a point of reference for use in comparative analysis. The decision-maker may ask qualitative or quantitative questions about effectiveness that this level of resolution may help explain. It is important to note that the analyst has not quantified the study question a great deal. The whole purpose of the analyst's decomposition to this point was to better define the study question, establish doctrinal and situational context, and create flexibility to later articulate the decision-makers interests. The analyst used the term "effective" in the sub-issue nodes in Figure 6. The term "effective" requires further quantification and grabs the attention of the analyst. The analyst must continue decomposing to define effectiveness in terms of the AH-64D's processes, tasks/activities, structures, and physical entity characteristics.

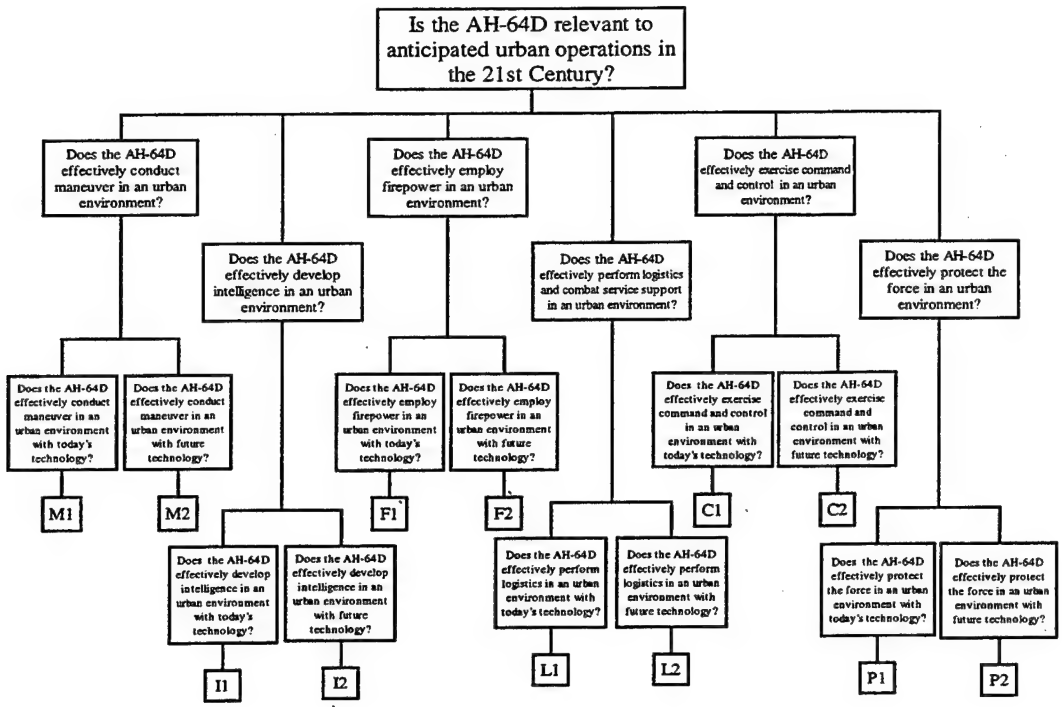


Figure 6. Study Question Decomposition to Today's versus Future Technology

(2) Decomposition to EEAs, MOEs, MOPs

The analyst can use tools such as lower level doctrinal references, the AUTL, and knowledge from the field to assist in further decomposition of the sub-issues into EEAs, MOEs, and MOPs. Since the purpose of the thesis is to describe and illustrate the SQM process, an exhaustive decomposition of the entire AH-64D study question tree will not be completed. For illustrative purposes, the thesis focuses further decompositional steps on sub-issues F1 and F2 of Figure 6. The other sub-issues identified in Figure 6 will not be decomposed.

The analyst continues the decomposition along the "Today's technology, Firepower" branch of the tree (F1, Figure 6). The analyst can address the processes, tasks, and activities of the AH-64D as it performs the parent functions identified by the

tree. The analyst uses the AUTL and FM1-140, *Helicopter Gunnery*, to decompose direct fire effectiveness (F1) into EEAs that address the elemental processes, tasks and activities associated with direct fire. These EEAs include detection, identification and engagement of targets, providing battle damage assessment (BDA), and the actual effectiveness of direct fire weapons in urban areas (Figure 7). [18] "Effectiveness" at this level of abstraction remains unquantified. The analyst must continue to decompose the EEAs by identifying particular detection, identification and engagement, BDA, and weapon effectiveness MOEs/MOPs. He or she may decompose these EEAs into MOEs/MOPs that continue to drive closer to data collection requirements.

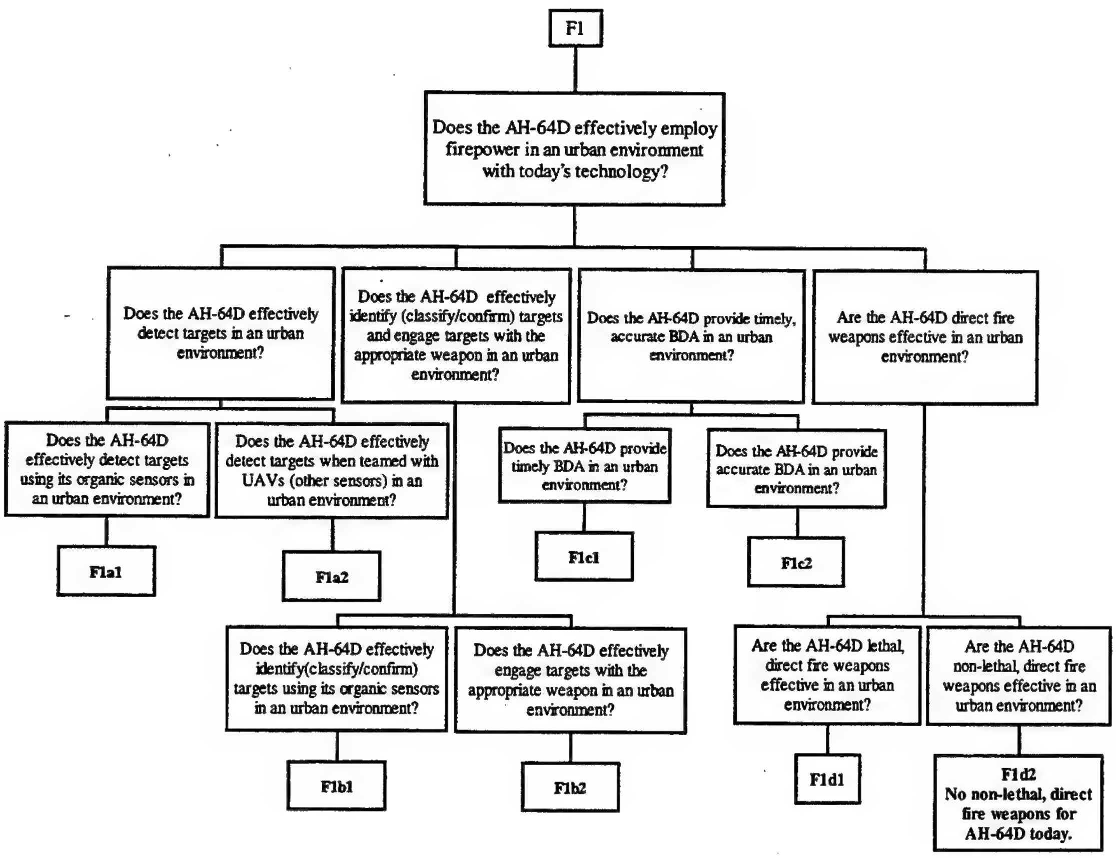


Figure 7. Employing Firepower in Urban Environments Today

The EEA for detection of targets in urban areas addresses the AH-64Ds organic detection capabilities as well as its detection capabilities when teamed with unmanned aerial vehicles (UAV). This focus may interest the analyst because of recent advances in UAV technology. The decision-maker may show interest in tactically employing UAV/AH-64D teams to detect and engage targets. The analyst may address identification and appropriate weapon selection in urban environments by understanding the elements of employing firepower as outlined in the AUTL and relevant attack helicopter manuals. Again, the analyst in this illustration can break down the EEA into AH-64D effectiveness in using its organic sensors to identify (classify/confirm) and engage targets with appropriate munitions in an urban environment. The BDA EEA decomposes into both a timeliness and accuracy issues. Finally, the analyst may address the direct fire weapon effectiveness EEA by including the employment of both lethal and non-lethal weapon systems in urban environments. Including lethal and non-lethal direct fire weapons allows the analyst to assess the AH-64Ds role in both MOOTW and conventional scenarios. Again, doctrinal references like FM 1-140, *Helicopter Gunnery* [19], and FM 1-112, *Attack Helicopter Operations* [16] provide the analyst needed context and serve to focus analytical effort.

Continuing the decomposition, the analyst develops MOEs or MOPs to quantify the EEA parent nodes. Figure 8 below illustrates the decomposition of F1a1 (Figure 7) into five MOEs. One MOE investigates the time it takes the AH-64D to detect targets in urban environments. Another MOE measures the percentage of targets

detected by the AH-64D given that the targets were detectable by the system. A third uses an overall percentage of detected targets to identify the AH-64D contribution to target detection vice other collection assets. Another MOE measures the percentage of personnel and vehicle targets detected in particular building configurations and subterranean features by the AH-64D. The final MOE investigates the ability of the AH-64D to detect threatening air defense assets in an urban environment.

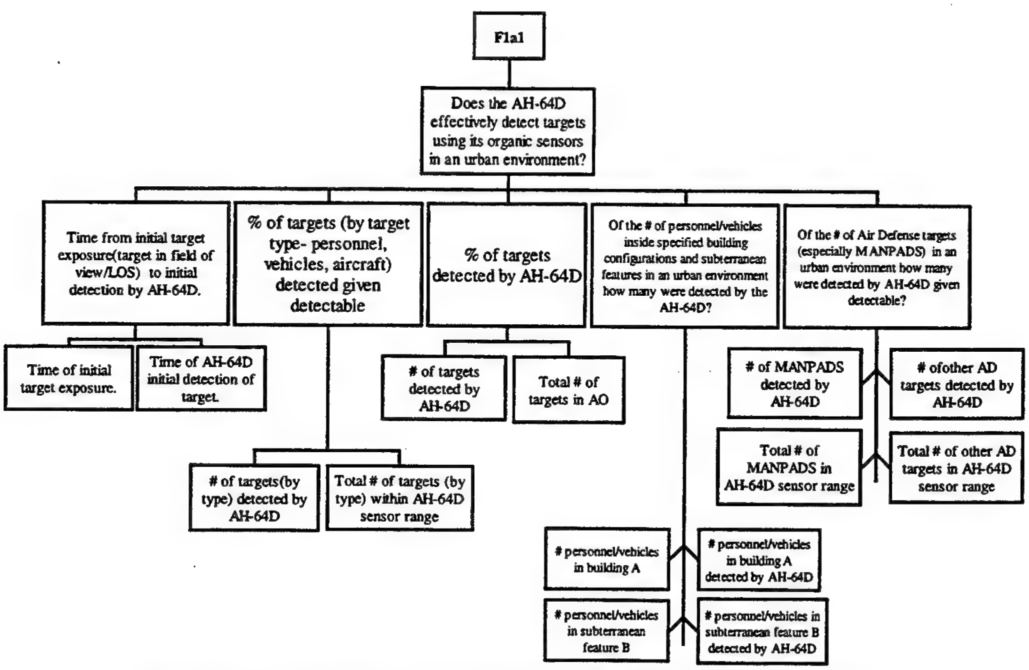


Figure 8. AH-64D Target Detection MOEs/MOPs Today

c. *Step Three: Identifying Data (Leaf Node) Level Requirements*

This stage of the process offers the opportunity to discuss how the analyst establishes important thresholds for determining the solution set of nodes. Analysts can develop the thresholds used by each node in the SQM tree to determine what the return value to the parent node will be. The analyst may identify these standards from doctrinal sources, experience, subject matter experts, and sometimes-even face-validity. Of

course, the analyst remains accountable and must articulate the rationale for any standard or threshold used in the process.

For example, the MOE that addresses the time from initial target exposure to the time the AH-64D detects the target (Figure 8) requires two input parameters. These input parameters (data (leaf) nodes) are initial detection time and initial target exposure time (see APPENDIX H for detailed discussion). The analyst must collect these times from a simulation, field experiment, or some other empirical source in order to compute the parent MOE. These two input parameters provide the parent MOE with the data it needs to calculate the difference in time. The node algorithm compares the difference in time to some user-defined threshold or standard to determine if the threshold was met. Depending on the threshold, a node answer is returned to the next higher level parent.

The analyst can use doctrinal references to help establish thresholds. These doctrinal references might not provide the precise measure the analyst requires. However, they may provide the analyst with sufficient information to develop enough of an insight into the process to allow the analyst to establish an appropriate standard. For example, the analyst needs to establish a detection threshold. The analyst may use FM 1-140, *Helicopter Gunnery* to extract a standard for use as the threshold for the MOE. The analyst will find FM 1-140 does not directly address the standard for organic sensor detection times during gunnery engagements. However, it does include an "Engagement Time Point Calculation Sheet" that can be extended to meet the analyst's. [19] This extension allows the analyst to include an algorithm and return type in the node. The

reader can refer to Section III.C.2 for the full-featured SQM details of how a node applies user-defined thresholds to calculate an answer to its question.

Now that the analyst completed the decomposition process for one branch of the tree, the analyst must decompose each additional branch of the tree. It is beyond the scope of this thesis to describe this full decomposition. However, the reader can refer to APPENDIX D for a full decomposition of the employing firepower, future technology branch (Figure 6, F2).

Figure 9 illustrates the analyst's decomposition of the future technology branch. It matches the tree developed for the current technology branch down to the EEAs but is modified by the future technology context. The analyst must decompose these EEAs into MOEs as previously discussed. This thesis continues the decomposition of Figure 9's F2d2 node in order to explore the potential non-lethal, direct fire weapons that may prove useful in MOOTW or special operation scenarios.

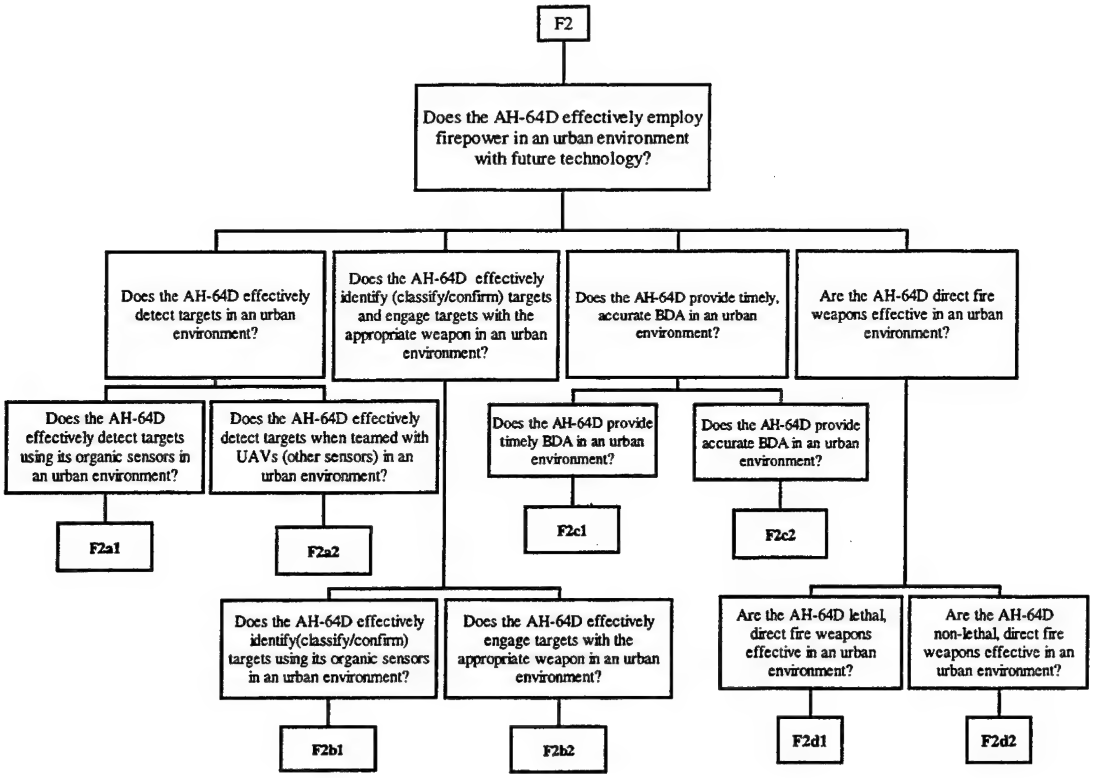


Figure 9. Employing Firepower in Urban Environments (Future)

The analyst's decomposition forces him to quantify the effectiveness of new "non-lethal" weapons. The reader can find a more detailed discussion of some of these non-lethal weapons in APPENDIX D. Some of the non-lethal weapons that the AH-64D could use in support of military operations in urban terrain include lasers using wide band illumination in the visual spectrum to cause temporary blindness. Additional non-lethal weapons include the AH-64D's use of remotely control UAVs to dispense irritant grenades, barriers, pyrotechnic irritant cartridges, or acoustic whistles. The AH-64D could help control crowds using rubber ball impact munitions, sting-ball, or flash-bang distraction rounds. Additionally, it could use its 30mm cannon to fire malodorant/irritant rounds. The AH-64D may use air-bursting munitions with

encapsulated sub-munitions like slippery foam and malodorants. Other direct fire, non-lethal weapons may include ground vehicle electric stoppers (GVES) mounted on the AH-64D that disable electronic engine controls [15] The analyst's decomposition becomes more challenging as he grapples with the challenge of quantifying some of these weapons' effects. The analyst may functionally decompose the F2d2 node as depicted in Figure 10. The analyst must decompose the remaining branches of the SQM in a similar fashion. This further decomposition is beyond the scope of this thesis.

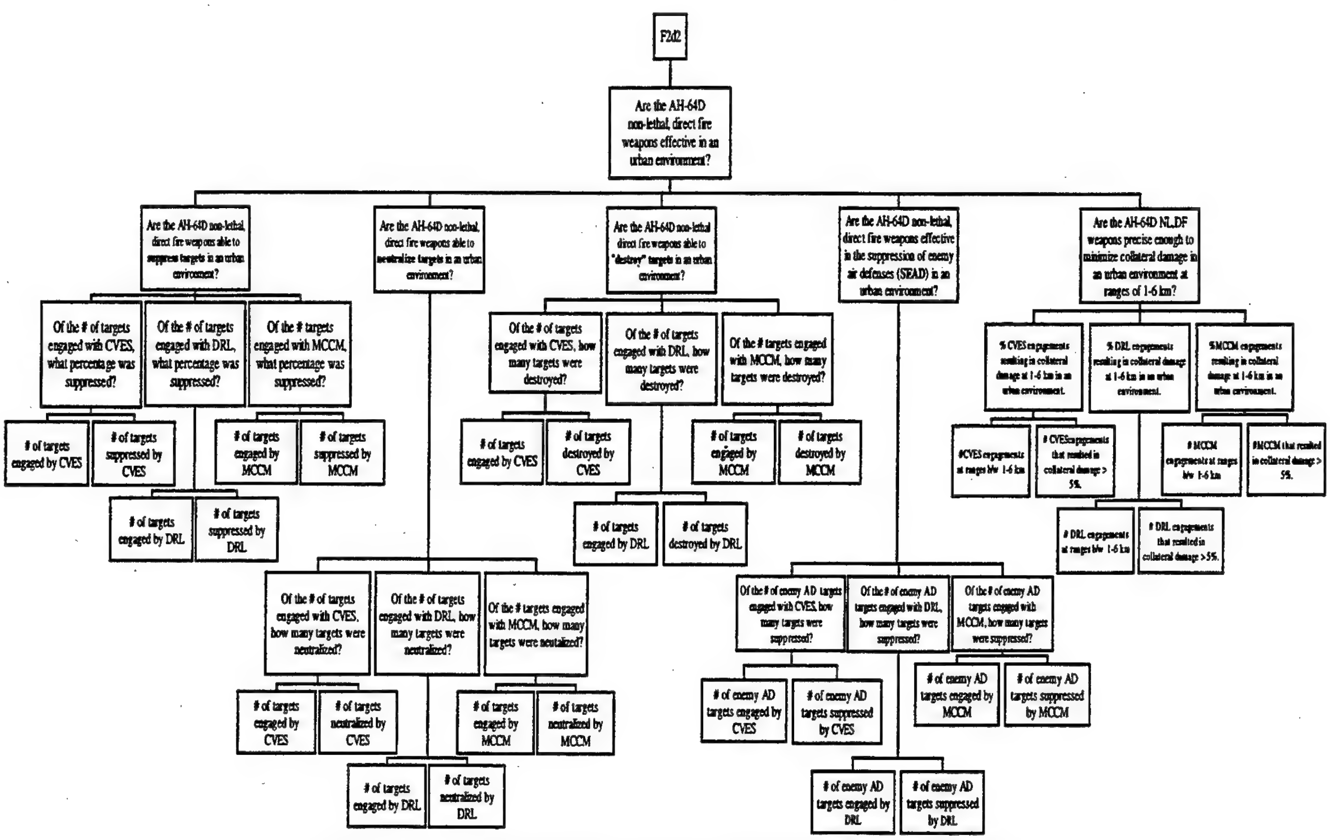


Figure 10. AH-64D Direct Fire Effectiveness (Future)

d. Step Four: Synthesis

First, the analyst must confirm that the data level requirements are collectable and answer all of the nodes in the SQM tree. Then, the analyst must synthesize from the deepest child back to the root node, to answer the study question. This thesis does not include real data collected during an actual simulation experiment and cannot show an empirical example of this step. However, all the leaf nodes in Figure 11 contain observable data that can be obtained from the federation. The following section does identify the process in sufficient detail to provide the reader an understanding of the requirements.

2. Full-Featured SQM Example

The SQM decompositional process described in previous sections identified the need to concurrently establish the node question, node algorithm, and node parameters for each node in the SQM tree (see Sections II.B and III.C). This section provides an example that illustrates in detail how to apply this process to one node. The overall decomposition steps described in previous sections remain the same.

This section describes a node characterization process for the SQM tree depicted in Figure 5. Each of the nodes in the SQM tree must be able to handle the four types of data previously mentioned (enumeration, Boolean, integer, and double).

The node algorithm in Figure 11 illustrates pseudo-code that an analyst could use to identify all of the important node characteristics of this portion of the SQM tree. The analyst looks for mappings between the qualitative words and quantitative data the question requires. In this illustration, the analyst posed six questions, each of which used

the term "effectiveness" as a measure of a particular subject of interest. The analyst developed these node characterizations during decomposition and may have refined them during synthesis.

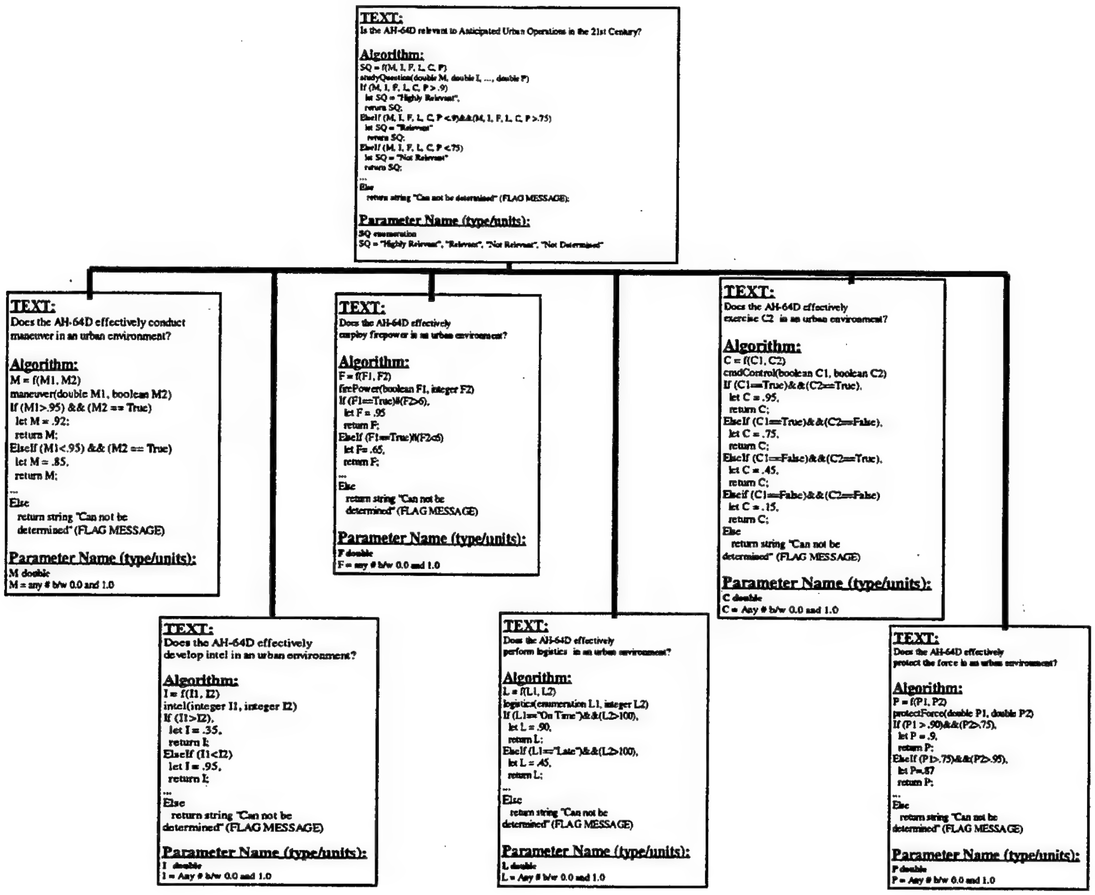


Figure 11. Node Characterization Pseudo-Code

Once all of the children of a particular node are populated with data, the parent node possesses all of the required inputs needed to execute its node algorithm and derive the answer for the parent node. In this example, the inputs to the parent are all doubles, as illustrated in Figure 11.

In this example, the analyst identified that the node parameter type would be the enumeration "Highly Relevant, Relevant, Not Relevant, or unknown." In order to arrive at one of these node answers, the analyst had to establish a node algorithm that would map the six child (parameter) inputs into one or some of these enumerated values. The values used in the pseudo-code were established based on his doctrinal decomposition and quantification.

The SQ node algorithm executes based on a function of its inputs (M, I, F, L, C, P). SQ is a function of maneuver (M), intelligence (I), firepower (F), logistics (L), C2 (C), and protecting the force (P) As synthesis occurs, parent nodes execute the algorithms necessary to arrive at an answer to the parent node. Here Firepower, F, is a function of today's firepower effectiveness (F1, a boolean) and future firepower effectiveness (F2, an integer) (Figure 11). Because SQ is an enumeration, the analyst must identify the combinations of inputs required to classify the SQ answer as "Highly Effective", "Effective", "Not Effective", "Not Determined." This is the point at which the analyst must use and establish thresholds to create a solution set for the node.

D. INSIGHTS

The application of the SQM to the AH-64D study question provided the author great insight into the process as well as illuminated some shortcomings in today's simulation models. As the SQM was applied down to data level nodes, the author realized that the data fidelity available from today's legacy models is inadequate for some of the future, non-lethal weapons concepts that need to be studied. As we look to the 21st Century, model developers need to address some of these shortcomings.

The characteristics of the urban environment discussed in Section III.A are not captured in the legacy models. Current aggregated models do not fight urban warfare in enough detail to allow analysts and decision-makers to do much more than speculate about results. Thus, the algorithms in the legacy aggregate and high-resolution models fail to capture the unique characteristics of urban warfare.

Today, there is a greater need to model weapons of mass destruction, chemical munitions, non-lethal weapons, and "unconventional" enemy forces. The simulation community has a long way to go before it can generate the data needed for analysis in the 21st Century.

The application of the SQM to the AH-64D study question illustrated the difficulty of conducting helicopter operations in and around urban environments. However, the application suggested that the AH-64D could greatly enhance military operations in support of MOOTW and other special operations. As applied, the methodology identified, and this thesis considered, sensors and direct fire sensing.

The sensors, in particular the FLIR, used by the AH-64D could be greatly affected by the physical characteristics and effects of the urban environment. Some of these effects include thermal clutter, high concentrations of carbon dioxide, urban haze, fires, smoke, rubble, and inclement weather. Although not absent in other environments, urban environments intensify or concentrate the effects. Urban areas are typically hotter than non-built up areas, since man-made structures retain a great deal of heat, making thermal discrimination more difficult. Laser designation is also less effective because of reflection from surfaces such as glass. The higher carbon dioxide concentrations also

hinder the transmission of thermal imaging. Distributed simulations could provide insight into these issues provided they possessed a high degree of fidelity to answer EEAs in these areas.

Employment of direct fire, lethal and non-lethal weapons is more difficult in urban areas. Depending on the nature of the urban landscape, the standoff advantage of long-distance missiles like the Hellfire is greatly reduced, negating this benefit when the helicopter is operated within the urban landscape. Other direct fire weapons like the 30mm or rockets may prove more useful but may lack the precision offered by smart missiles. The 30mm offers greater accuracy than the rockets. Special munition improvements in rockets and ammunition for the 30mm, particularly in non-lethal variants, may make the AH-64D indispensable in MOOTW and special operations (combat search and rescue (CSAR), raids, etc.).

Sensing and engagement times are shorter in an urban environment than conventional environments because of the ease with which targets enter and exit dead-space or blind spots. Buildings, subways, rubble, and other ground clutter allow dismounted enemy forces to move rapidly and minimize their expose to the AH-64D's sensors. Future sensors that can see through some of this clutter may greatly enhance the AH-64D's role in urban environments in the next century provided improvements in the survivability of the aircraft are also considered.

One area that requires further research is that of protecting the crew and AH-64D from the effects of close combat. If the expectation is that the AH-64D will navigate through and around urban canyons in MOOTW, conventional, or special operations, it

may need greater protection than the 12.7 mm currently provided. The nature of urban canyons suggests that the survivability of the AH-64D will depend on its ability to mask itself from identified threats by using the buildings and structures of urban environments. Battlefield omniscience becomes even more important for those considering sending AH-64D crews into an urban environment. Teaming AH-64Ds with expendable UAVs seems a much-preferred method of employment to protect AH-64D crew and its equipment from a rather nasty environment. As the size of UAVs decreases and the targeting systems become more advanced, AH-64Ds pilots may be able to control UAVs much like mounted infantry forces use dismounted troops. Miniature UAVs can be used like dismounted troops to clear or secure areas into which the controlling force wants to move.

This chapter demonstrated the application of the SQM to a study question and illustrated its decompositional steps. The chapter provided the context necessary to discuss automation implications for an ASQM. The ASQM is the subject of the next chapter.

IV. AUTOMATED SQM IN HLA

This chapter describes the automation implications and architecture requirements of the SQM necessary to make the ASQM. Section A discusses the design precepts for the ASQM. Section B proposes a method by which to populate the ASQM tree during distributed simulations and provides the key link that makes the SQM a fully automated tool (ASQM). This chapter concludes with Section C that applies the ASQM to the AH-64D study question decomposition to demonstrate required functionality for the automated tool.

A. AUTOMATED SQM DESIGN PRECEPTS

An automated implementation of the ASQM must provide the analyst with a user-friendly graphical user interface (GUI) that insulates him from programming details associated with collecting and analyzing data in distributed HLA simulation sessions. The tool must automate the implementation of the HLA service invocations. This requires a conceptual framework that supports the automatic execution of HLA functionality that is traditionally implemented in simulation specific computer code. Subscription and publication are examples of the services that must be provided to the analyst via the GUI interface. These subscription and publication capabilities must be easily adapted for use with any FOM via the GUI. There should be no associated requirement to rewrite code to subscribe to or publish data. The goals for the ASQM must include the ability to traverse the native FOM to select base data elements needed to populate the SQM tree. It should include automatic subscription procedures, populating

the SQM database with native FOM (object-oriented database) forms, and allowing for graphical interactivity. The ASQM should facilitate pre-exercise, real-time, and post-processing analysis.

The tool that implements the ASQM must possess functionality transparent to the user. Analysts with little familiarity with object-oriented programming should be able to use user-defined terms to build their SQM tree. The GUI should provide a point and click interface to allow the mapping between the user-defined data (leaf) nodes and the FOM objects. The Analysis Federate [8] provides the subscription functionality that the ASQM can extend to allow users to map federation objects to required data elements.

B. AUTOMATION PROPOSAL

The ASQM relies on data from the federation. Getting the needed data to the nodes of the SQM tree is not a trivial task because the current HLA design precepts do not address how this is best accomplished. This thesis proposes one method for populating the SQM tree with data. This proposal addresses the architecture required to fully automate the SQM and serves as one method to reach the desired end-state of a fully ASQM.

When the analyst develops the SQM tree, he or she knows the structure of the FOM federation's specification, that is which attributes, interactions, and parameters are in the federation, and thus available for collection. The major issue is how the data nodes get the required data from the federation.

The federation RTI publishes to each local federate RTI in byte streams in accordance with the FOM specifications. The local RTI then interprets and converts the

byte streams to the federate's SOM so the federate can then use this information internally.

The SQM can be fully automated by applying a data-filtering methodology that was proposed by Murphy and Aswegan [20] to address RTI inefficiency. During HLA distributed simulations sessions, the architecture forces federates to subscribe to and receive interaction and attribute information updates continuously to meet data needs that may only require periodic or conditional updating. [20]

Murphy and Aswegan's data-filtering approach called for extensions to the HLA standard that would allow federates to remotely execute their filtering logic in the local RTI components of other federates. This remote logic execution would reduce network bandwidth requirements by facilitating information filtering by a potential data producer. [20] This approach recognizes that the simulation that is consuming data is the only distributed simulation component that is capable of accurately specifying what types and under what circumstances it requires data. It places the entire overhead associated with specifying data filtering requirements on the data consumer. [20]

This thesis applies the concept of "remote data filtering" between federates to automating the SQM within an HLA federation. This thesis proposes that the ASQM can "remote" its algorithms to another process that conducts queries into a data base to get required data. For further discussion of how this thesis uses the concept of remote data filtering, see APPENDIX E.

Recall that the Analysis Federate is an important tool that insulates the user from the overhead associated with developing HLA federates. The automation techniques that

were introduced in the Analysis Federate research can be capitalized on to provide analysts with the ability to subscribe to data that their front-end analysis identifies while using an automated SQM (ASQM). An ASQM provides the front-end formalization and thoroughness analysts need when investigating a posed study question. An ASQM with the Analysis Federate serves as an important pre-execution tool to assist the user in determining what data they need to collect to support analysis. Both go further by allowing the user to identify the equations needed to use the data in a meaningful way, and the ability to populate the equations with parameters.

C. APPLIED ASQM SCENARIO

The following section describes the ASQM step-by-step and serves as a template from which software developers can explore the programming considerations of the ASQM analysis tool. The section highlights the major processes an analyst conducts as well as the functionality that should be available to the user at each step. The illustrations identify functionality issues and do not depict the only way to present the user with required user input. The GUIs presented do not depict the "approved solution", but merely suggest the functionality that must exist in the ASQM in order for it to prove useful to analysts.

1. Working Node GUI

The following illustrations depict the functionality required at any node within the ASQM tree. Individually, these GUIs may suffice in a final ASQM software package. Throughout the process, a graphical representation of the ASQM tree, in whole or a useable portion of the tree, should be available to the user. As the analyst applies the

ASQM, he or she proceeds through the logical sequence of steps as described in Chapters II and III. The ASQM must capture this logical sequence and provide the user a practical method by which to enter required information.

a. Node Question Interface

The first step of the ASQM process always begins with user identification of the study question. In the attack helicopter example, the user first enters the study question in text format (Figure 12). The interface only requires the user to type in text. The user decides how much "formalism" he or she requires by entering any text entry that sufficiently illustrates their ASQM tree. This Node Question Interface serves as the interface for any particular node within the tree. When the user clicks on a node from a graphical representation of the tree, he or she should be provided this interface or have the option to view this information. This interface allows the user to identify the node question that is the first of three required node characteristics (see Section II.B.2.b).

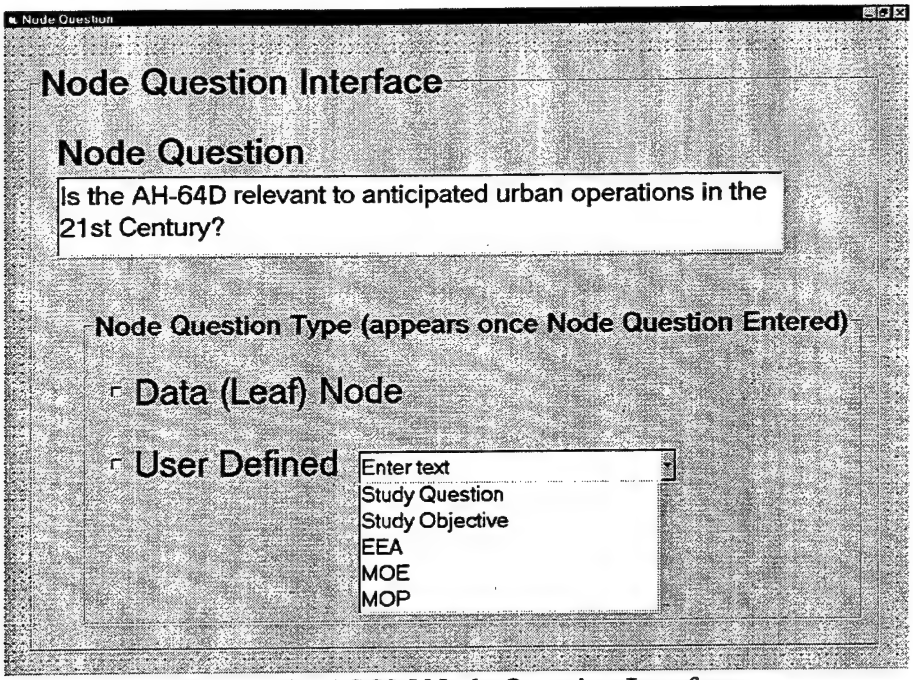


Figure 12. ASQM Node Question Interface

Once the user enters the question, a dialog appears that asks the user to identify the node type (see Figure 12). Identifying the node type allows the analyst to print or view only those nodes in which he or she is interested. For example, the analyst may want to see all of the MOE or data (leaf) nodes of the tree. This type of information could be used to display SQM tree information like the number of nodes and the tree depth. It requires the user to identify data (leaf) nodes in particular because that node type initiates critical actions such as subscription to HLA objects and interactions. A separate GUI, described later in this section, illustrates the functionality required once the data (leaf) node is checked. Once the user enters this node information, the ASQM automatically leads the user to the next step in the process.

b. Node Answer Data Type Interface

The process needs two essential pieces of information about the current, working node to provide functionality to the ASQM. The user must identify both the anticipated answer type of the node and an initial (default) value. The analyst selects what he or she expects the answer of the current node will be as well as a default setting for the node. This helps prevent run-time errors.

The user establishes the Answer Set Data Type by a dialog shown in Figure 13. This information is required because the user should know what the possible values of the node will be once the node uses its children's data in its algorithm to arrive at an answer. In the AH-64D application, the analyst selected enumeration as the solution set for the study question. If the user selects Boolean, the answer defines either two states "true or false" or "yes or no." When selecting integer, the answer will be integer in type. Partitioning into a finite number of subsets occurs at a later step. When selecting double, this same type of finite partitioning also occurs.

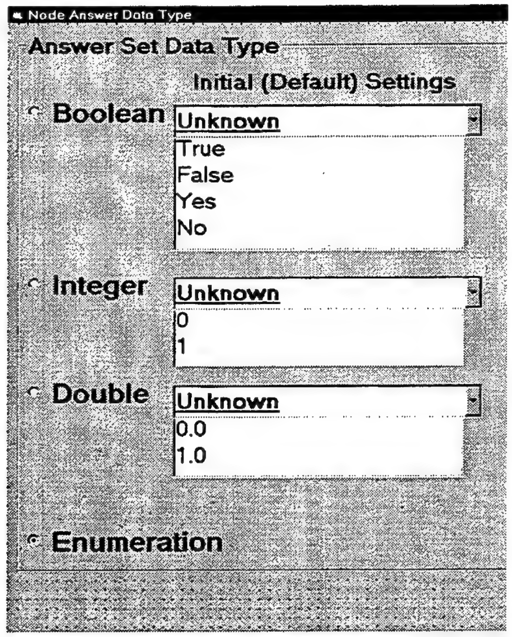


Figure 13. Node Answer Data Type Interface

The user can establish the initial (default) setting for the node (Figure 13). This initial setting serves many purposes. The ASQM uses the initial setting to ensure the entire answer set population is complete. When the user later creates a mapping function for the working node (Figure 18), he or she knows that any interaction not captured by the user's function will map into the default setting established at this point in the process. More precisely, the default setting exhausts the domain and allows the ASQM to provide answers to parent nodes (see APPENDIX F for discussion). Default settings prevent run-time errors by always returning a value to a parent. With the return type of unknown, the ASQM can flag a message to the user stating that some data could not be collected, was corrupted, or state the nature of the error. The ASQM software should be functional enough to return a qualified answer for that particular node.

The ASQM recommends initial setting of "Unknown" to the user for all data types. This ensures that the parent's algorithm, if it has a parent, will not return an error message to the user without specifying a reason. Boolean, enumerated, double, and integer types each have logical default settings that the user should be aware of when constructing the tree (prompt with possible settings like those in Figure 13). "Unknown" initial settings provide the user the flexibility to define later (when more information is available) the criteria by which to judge data inputs. It has an indirect effect of indicating that the node is not as restrictive and may not be as important as another, differently defaulted node. Clearly, the user is the only one who can articulate relative importance based on his or her decomposition.

When the user selects enumeration in Figure 13, a dialog such as Figure 14 appears. This example illustrates the user's entry for the AH-64D SQM application. The analyst decided to describe the answer to the study question with four possible responses ranging from "Highly Relevant" to "Unknown." Unknown should always be included as a possible return type for reasons previously discussed. Additional ASQM functionality should include the ability to select from previously used enumerations or a pull down menu of other options. Standard enumerations organizations use to articulate relative scales include the Army's T,P,U (trained, practiced, untrained) standards for all METL tasks.

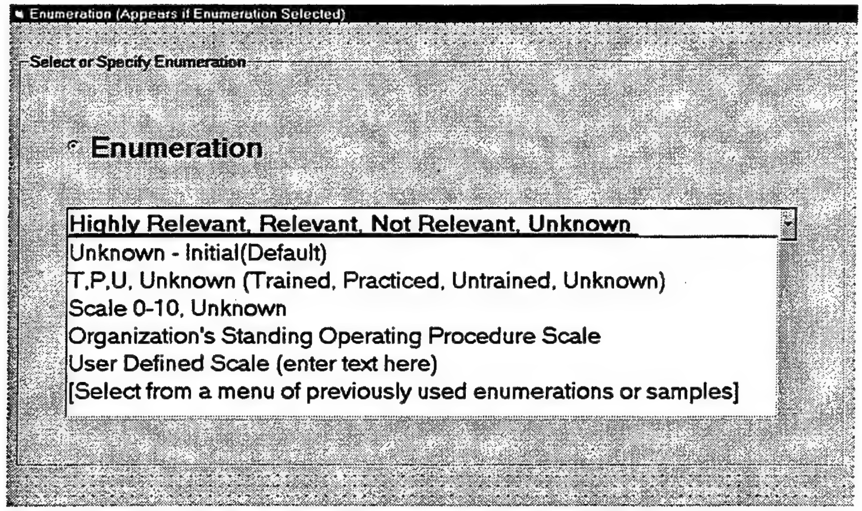


Figure 14. Enumeration Interface

2. Node Parameters and Mapping

Once the user establishes the answer set data type and defaults, the ASQM prompts the user to identify the number of child nodes, the data types of those children, and the appropriate range of those types (Figure 15). This section describes interfaces that the ASQM uses to decompose the working node into its parameters (child nodes). It further identifies how the parameters combine to produce an answer set for the working node. The user enters information about each child and establishes a mapping function to describe how the child node inputs combine to produce an answer to the working node. This section's interfaces allows the user to identify the node parameters and algorithms that are the third and second of three required node characteristics, respectively (see Section II.B.2.b).

a. Child Node Interface

Figure 15 depicts an ASQM interface to specify children nodes. Each child node serves as a parameter to the solution algorithm of the parent node. In the AH-64D SQM tree, six sub-issues or questions served as children to the root study question. The user enters the number of children to the working node or selects "Add Child Node" from a pull down menu. Users first enter text describing the first child node in the same manner done for the node question GUI. This entry creates a node reachable from this interface or the Node Question Interface (Figure 12). This child node information should be reflected to other GUIs that use the same information. He or she can then point and click on any node from a picture of the current tree and begin working on it and its current information is displayed. Next, the user identifies the child node return type (Boolean, enumeration, double, or integer) based on the question (Figure 15). The user enters the domain of the child node (or updates automatically based on what the user defined as a return type). If the user has not fully defined the domain, he or she can do so later when working on that particular node.

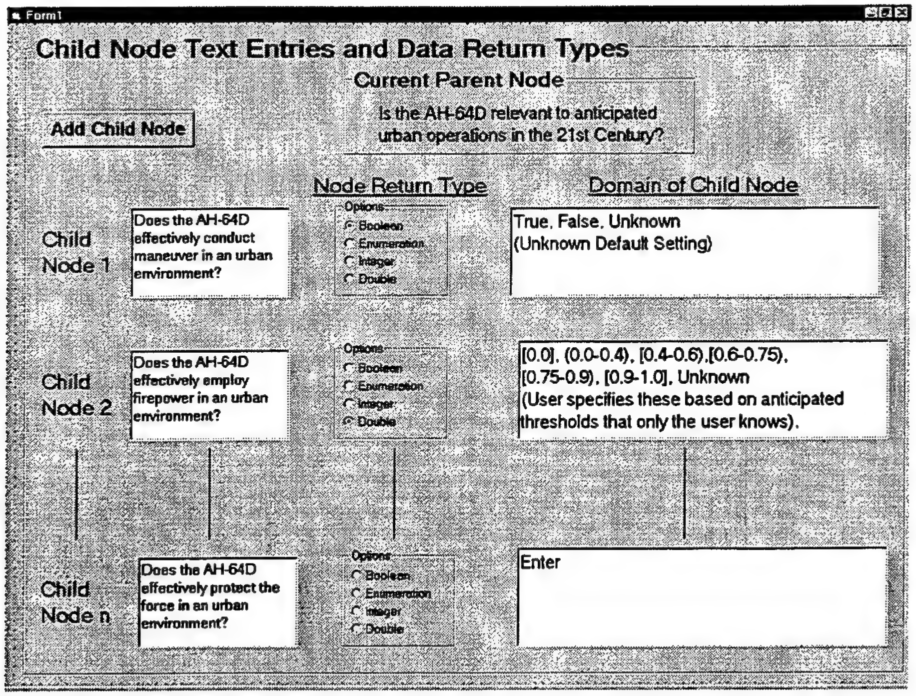


Figure 15. Child Node Input and Data Return Types Interface

b. Node Algorithm Interface

The user is now ready to define the mapping function of the current working node. This function could be a weighted average, an average, or any function the user wants to define. The user should be able to select from a drop down menu or pallet a particular algorithm or function (statistical, math, financial, etc.) that he will use to combine the child node answer possibilities into the parent node answer.

Other ASQM functionality should include a spreadsheet-like interface that allows the user to define the working node mapping function as shown in Figure 16. Using the AH-64D application with two children (top matrix of Figure 16), the user could identify how the combinations create an answer to the working node question. During run-time, the algorithm of the working node would take inputs from the distributed

simulation and calculate an answer to the question based on the mapping established by the analyst in this matrix. Here, the user only identifies the relationships he or she feels are important and lets the earlier established default settings map other permutations or combinations into unknown status to prevent any run-time errors. The user edits an interaction matrix as shown (spreadsheet like functionality) or just points and clicks from menus of child node objects. The matrix at the top of Figure 16 illustrates a simple case of two children; however, the ASQM must possess the functionality to allow the user to provide input into innumerable combinations of children. A pivot table provides this functionality (Figure 16, bottom illustration). The user should be able to address the domain of the node answer set completely and exhaustively by articulating results for only the combinations of interest. The software should use the default setting identified earlier to map any unspecified combinations.

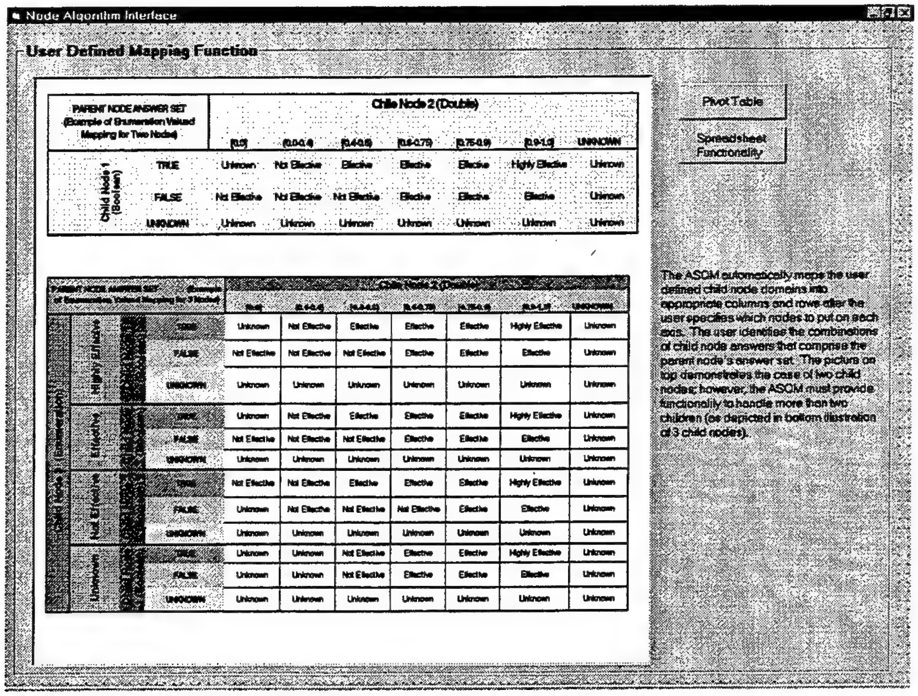


Figure 16. Node Algorithm Interface

A pivot table is a highly effective method (although not the only one) by which to discretize the solution set. It allows the user to organize large amounts of data by putting the source data into columns and rows. The pivot table provides summary, cross-referencing, and filtering functionality and allows the user to enumerate combinations of child nodes. Most nodes with an enumerated answer set will not have more than a handful of children. If any one node has more than 8 children, then the analyst may not have decomposed the study sufficiently. Functions in nodes that require numerous children as parameters will not be as complicated as the case for enumerated values. Engineering level applications may prove to be heavily reliant on multiple children to answer the parent node. In these cases, the analyst probably does not need to enumerate a solution set. If an analyst using the pivot table finds that enumerating 15

combinations of parameters is too confusing, perhaps the analyst can return to the parent node or "grandparent" node and try another decomposition that provides a sufficient answer.

3. Data (Leaf) Node Subscription

When the user identifies the working node as a data (leaf) node (Figure 12), the ASQM must prompt the user to subscribe to the appropriate FOM object (Figure 17). The selection of a data (leaf) node serves as a cue both to the ASQM programmer and user that the critical connection between the ASQM and the distributed simulations is about to occur. Through the functionality suggested by the following interfaces, the ASQM can meet the needs of users by merging the unique front-end analytical process and subscription requirements of HLA.

a. Navigate FOM/SOM Interface

The FOM and SOMs must be available to the user throughout the process in order to identify information to which he or she must subscribe to populate a particular data node. As depicted, the DMSO distributed Object Model Development Tool (OMDT) software provides the user all required HLA information about the FOM/SOM. The software tool developed in support of the ASQM must provide similar navigation functionality. Once the user navigates this FOM/SOM information using the ASQM GUI he or she can begin subscription to required data (attributes, interactions, etc). Figure 17 shows a point and click, drag and drop interface that allows the user to map between the analyst's terminology and the FOM objects.

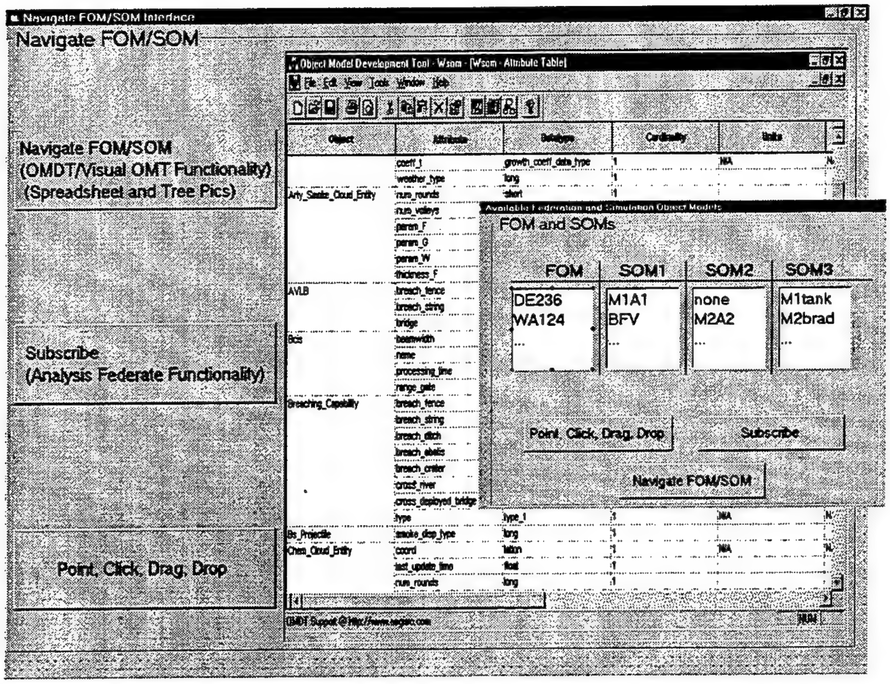


Figure 17. Navigate FOM/SOM GUI

b. Subscription/Mapping Interface

Once a user selects an attribute or interaction from the GUI in Figure 17, that object's information displays in the subscription/mapping dialog of Figure 18. The user can then map between the ASQM tree and the FOM objects. The ASQM should display FOM and ASQM Tree names (not just nomenclature as depicted) so the user would not have to recall the conventions used by the software. Again, a graphical representation of the ASQM tree can help cue the user about which node is the current data node. The user points and clicks on the desired FOM object to map to a particular node. Once all the nodes are mapped the ASQM can be used to automatically join a federation.

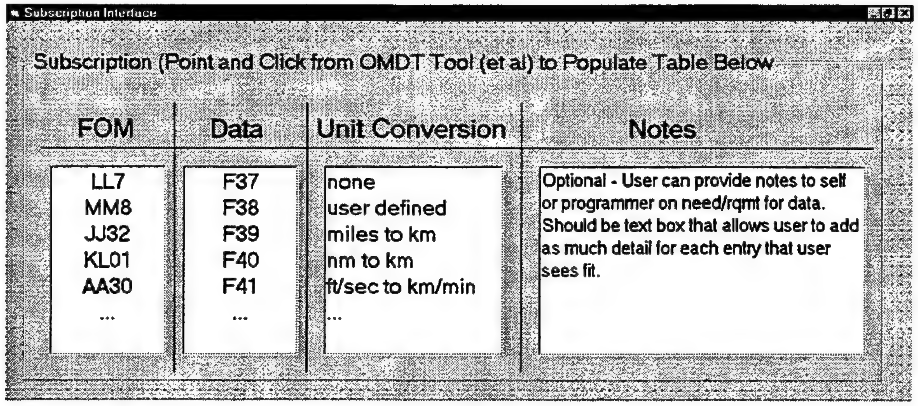


Figure 18. Subscription/Mapping Interface

Additional functionality should include the ability to do unit conversions if the federation publishes in different units than those needed by the ASQM tree. Casting functionality may also be appropriate now for unique mapping needs. Finally, the user should be able to write notes for himself and others as to why particular objects were subscribed to as well as serve as information for programmer's should their services be needed during troubleshooting.

4. ASQM Additional Functionality

a. Pallet of Automation Tools

Throughout this process, a pallet of automation tools must be available to the user. Figure 19 depicts command buttons of functionality that must be available so that the user can ensure the tree is simply constructed and easy to use. Some examples include fidelity checks that ensure the number of parameters equals the number of child nodes. This helps to ensure the entire answer set for the working node is exhausted/descretized. The ability to navigate the FOM objects, attributes, and interactions is another functionality that the user requires. The user can add or delete

nodes, and convert units as required. Functionality must include reuse of functions and applicable branches of trees from internal and external libraries, and the ability to cast from one data type to another should that be necessary.

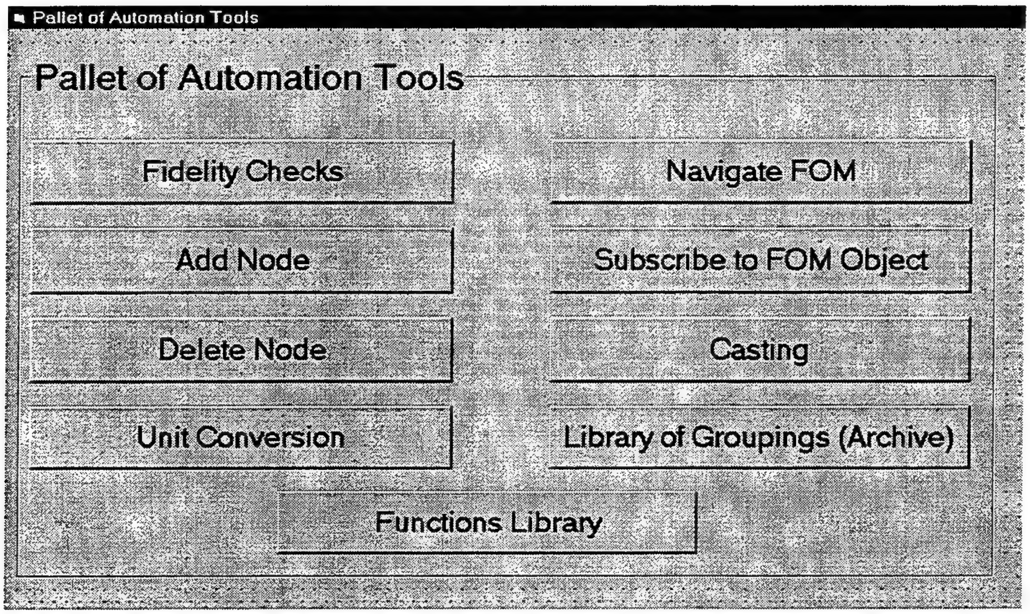


Figure 19. Pallet of Automation Tools

b. Post-Processing Tools

The preceding sections described ASQM functionality requirements. This section shifts the focus from the ASQM to the separate analysis tool that allows the ASQM user to generate arbitrary queries to the database (see APPENDIX E, Figure 25, P4). Recall that an analysis tool (P4) possesses the ability to display the ASQM results in real-time, but it does much more. The user should be able to use the analysis tool (P4) GUI and existing functionality to conduct sensitivity analysis or alternate tree synthesis. This capability would allow the analyst to explore what nodes seemed to possess the most

influence as well as vary the results to see what impact changes have on the overall answer or any branch of the ASQM tree.

The analysis tool's potential functionality should include the ability to perform statistical analysis on data. Once the tree is complete and the study question answered, the analyst may wish to conduct a sensitivity analysis or identify nodes with exceptional influence. Changes to influential nodes may change the answer to the tree and would thus be of particular interest to decision-makers. By articulating these decision thresholds to the decision-maker along with the audit trail provided by the ASQM, the analyst could help the decision-maker better understand the risks under which the decision is made. By allowing the user to perform sensitivity analysis by varying the inputs or randomizing the data level entries based on the results of the simulation, the ASQM's functionality and usefulness increases. By selecting a "simulate" option from a pull-down menu or button, the analyst could generate Monte Carlo results for each node (based on Monte Carlo or the distribution the simulation dictated or user specified). This simulation allows the analyst to do sensitivity analysis on the results and see what the critical or most influential nodes are.

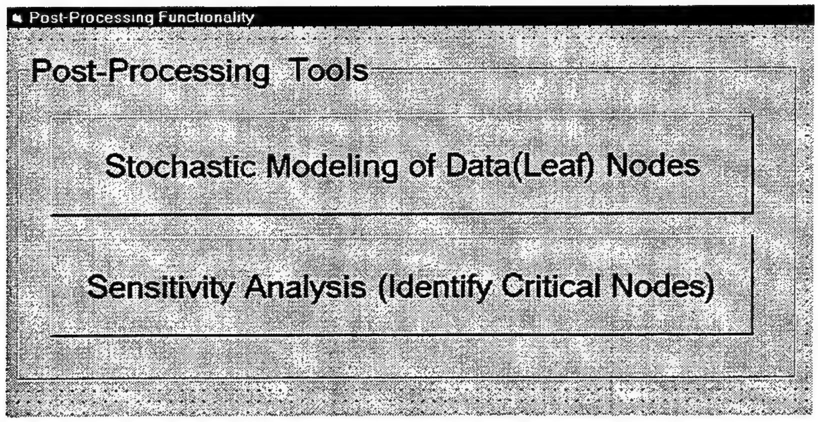


Figure 20. Post-Processing Tools

The ASQM described in the preceding sections provides the analyst the functionality required to decompose a study question into its elemental data. It also provides the ability to automatically link the ASQM tree to the data required to populate the tree and answer the study question. When implemented, the ASQM tool will help fill an analytical void that currently exists in HLA distributed simulations.

V. CONCLUSIONS AND SUMMARY

A. CONCLUSIONS

The study question methodology fills an analytical void in HLA. Analysts need the structure provided by the SQM for asking questions in the future because of the unknown nature of those questions today. Information, psychological, and urban warfare scenarios, coupled with future technologies, confuses the issue of study question development. Additionally, these questions drive model requirements. From the trained analyst's perspective, this hierarchy of needs or higher level of abstraction for which there are less predictable questions requires a process like the SQM.

The ASQM goes beyond what is possible with the SQM. The ASQM enables analysts to establish a clear, automated audit trail from study question to data requirements. It provides the analyst the flexibility to identify needed algorithms, subscribe to data that their front-end analysis identifies, and conduct real-time analysis. The ASQM meets an analytical need of the HLA.

While the ASQM has been developed in the context of HLA distributed simulations, it could be used for analysis of other simulations, distributed or stand-alone. For example, the ASQM can serve as a decision-support tool in that the user could build an ASQM tree but not automatically populate it with data. Data to populate an ASQM tree could come from manual observations or any source.

Potentially this tool can include other types of simulation, Internet applications, and methods by which to collect and analyze data. Both the manual and automated

application of the SQM remain applicable in areas such as AoA studies, materiel acquisition processes, and training effectiveness analysis. An ASQM used in HLA distributed simulations provides functionality not available with manual SQM applications.

Figure 21 illustrates the focus of this thesis as well as where subsequent research is required. This thesis was able to describe the manual pre-exercise processes and suggests a method that can be used to automate this process. Follow-on research is required once an ASQM is developed in order to exercise the software and confirm that it is functional and meets the analyst's needs.

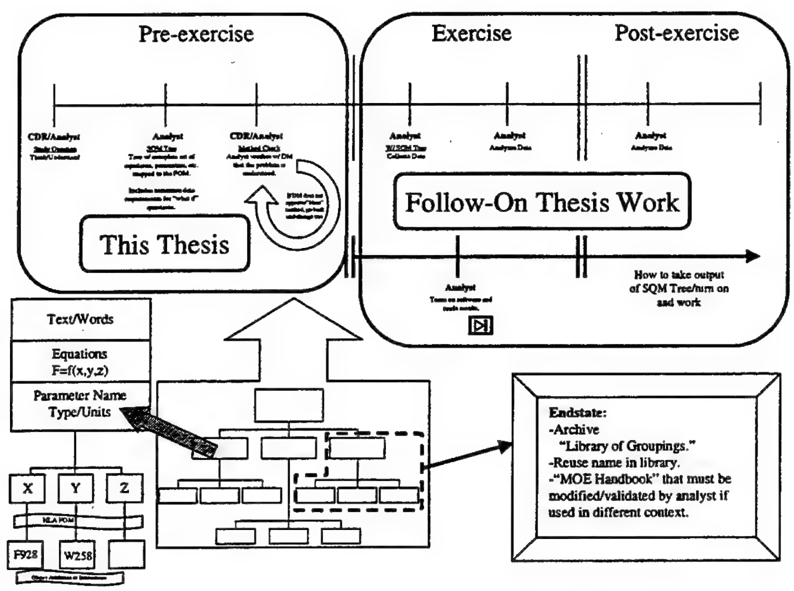


Figure 21. Conclusions and Future Work

B. SUMMARY

This thesis has developed the SQM, a dendritic approach that can be used by analysts use to identify MOEs, MOPs, and data requirements for studies and tests. It

demonstrated the general applicability of the SQM to answering study questions and provided an example of an Army specific application. This thesis used the lessons learned during the manual application of the SQM process to identify and develop an ASQM capable of meeting an analytical need in HLA distributed simulations. It provided the general solution for the ASQM that software developers can use as part of a requirements document.

APPENDIX A. GLOSSARY OF ACRONYMS

AD	Air defense
ART	Code for Army tactical tasks in AUTL
ASQM	Automated Study Question Methodology
AUTL	Army Universal Task List
BDA	Battle damage assessment
BOS	Battlefield Operating Systems
C2	Command and control
CAS	Close air support
CBRS	Concept Based Requirements System
CS	Combat support
CSAR	Combat search and rescue
CSS	Combat service support
DA	Department of the Army
DMSO	Defense Modeling and Simulation Office
DoD	Department of Defense
DT&E	Developmental test and evaluation
FM	Field manual (Army)
FORSCOM	Forces Command
GPS	Global positioning system
HQDA	Headquarters, Department of the Army
IFF	Identification, friend or foe
IPB	Intelligence preparation of the battlefield
IR	Information requirements
JCS	Joint Chiefs of Staff
JTF	Joint task force
LAN	Local area network
METL	Mission essential task list
MOOTW	Military operations other than war
MTOE	Modified table of organization and equipment
MTP	Mission training plan
NBC	Nuclear, biological, and chemical
NTC	National Training Center
OOTW	Operations other than war
OP	Code for operational-level tasks in AUTL and UJTL
OT&E	Operational test and evaluation
Pam	Pamphlet
ROE	Rules of engagement
SEAD	Suppression of enemy air defenses
SN	Code for national military strategic tasks in AUTL and UJTL
SOF	Special operations forces
SOP	Standard operating procedure
SQM	Study question methodology

ST	Code for theater strategic tasks in UJTL and AUTL
TBP	to be published
TM	Technical manual
TOE	Table of organization and equipment
TRADOC	Training and Doctrine Command
TTP	Tactics, techniques, and procedures
UAV	Unmanned aerial vehicle
UCOFT	Unit Conduct of Fire Trainer
UJTL	Universal Joint Task List
U.S.	United States
WAN	Wide Area Network
WMD	Weapons of Mass Destruction
WW	World War

APPENDIX B. DISTRIBUTED MODELING AND SIMULATION

1. General

As the Department of Defense (DoD) continues to look for ways in which to sustain its readiness and minimize costs, it faces the daunting task of training its forces for current operations and anticipating future operational requirements. Analysts primarily focused on large scale, European based scenarios before the fall of the Berlin Wall and collapse of the Soviet Union. These events, combined with less stability in other parts of the world, have caused the military to shift its focus to "non-traditional" missions. This change in focus is a paradigm shift [21] and requires a significant investment of time and resources for analysts to transition their methodological approach.

With the advent of the modern computer, the DoD began to synthetically model military interactions involving its people and systems. These simulation models provided the military a cost-effective method by which to train the force, investigate system acquisition and force structure issues, and analyze operational alternatives. Each service or community within the DoD focused on its own simulation requirements using its own standards. Thus, the DoD modeling community could not combine models to replicate higher order, joint systems.

As the nature of simulations and technology changed, the DoD pursued ways in which to combine the best models from each of its communities to simulate larger systems. To do this required a common standard and simulation support environment capable of combining models to meet user's needs. The DoD concluded that establishing a common simulation support environment would support model reuse, reduce overhead

and waste, and permit greater fidelity for each DoD component. Establishing such a general distributed simulation support environment permitted the DoD to enhance its use of simulations to maintain and improve its war-fighting readiness.

2. Types of Military Simulations

There are three types of simulation used in the DoD: virtual, live, and constructive. Virtual simulations use real people using simulated systems operating in a synthetic environment. The Army uses the Unit Conduct of Fire Trainer (UCOFT) virtual simulator to train its M1 Tank and M2 Bradley Fighting Vehicle (BFV) crews before and during field gunnery. The UCOFT demonstrates the usefulness of virtual simulations. Its helps improve the crew's coordination and overall proficiency. It also conserves ammunition by reducing the number of qualification repetitions required for each crew during gunnery live-firing training. Live simulations use real people using real systems operating in a real environment. Army units participate in live simulations during rotations to the National Training Center (NTC) at Fort Irwin, California, live fire exercises conducted on local training area ranges, and deployment exercises that test a unit's ability to deploy. Constructive simulations use simulated people using simulated systems operating in a synthetic environment. Army examples of constructive simulations include JANUS, ModSAF, and numerous other simulations used to train units and test concepts in the pursuit of readiness. [4]

The DoD uses virtual, live, and constructive simulations for many purposes, the most important of which is maintaining and improving war-fighting readiness. For example, the Army uses virtual, live, and constructive simulations to train its soldiers,

test and evaluate its equipment, and explore new war-fighting concepts in support of DoD readiness. The Army and other DoD organizations rely on all three types of simulation to enhance their war-fighting capabilities.

Simulations used by the Army training community are either high or low-resolution models. Low-resolution models aggregate combat units at brigade and higher levels. High-resolution models focus on details at levels below brigade. [4] The appropriate level of resolution for a particular simulation session is determined based on the study question, available time, analysis requirements, and hardware and software issues. For example, The Army's Training and Doctrine Command Analysis Center at White Sands Missile Range (TRAC-WSMR) is responsible for supporting brigade sized units and below. They use high-resolution models to investigate issues at battalion, company, platoon, and squad level. Similarly, their counterparts at Fort Leavenworth (TRAC-FLVN) are responsible for supporting division and corps units. They use low-resolution to test concepts and help train higher level staffs. These simulations provide insight into war-fighting issues at different echelons.

3. Distributed Modeling and Simulation

Early simulations and computer models consisted of self-contained processes. These early simulations were referred to as stand-alone systems because they did not interact with other computers. Rather, they fought scenarios internally. These early models were quite cumbersome because all algorithms and mathematical models were internal components of the stand-alone system. This approach limited their analytical use to a narrow, pre-determined focus.

The computer based simulation support environments that emerged during the late 1970's facilitated the concept of distributed simulations. [22] Distributed simulations use computer networking technology to allow individual models to communicate with each other. This allows a simulation system to be composed of numerous components or sub-systems that interact with each other over a computer network. The military's Simulation Networking (SIMNET) project demonstrated that distributed models could effectively interact. [4]

The attractiveness of composing simulations from model components is that it allows the freedom to use models in the same manner as stereo components. Stereo users who use the component approach have the ability to select particular components based on their preferences for high quality sound processing, price, functionality, and other factors. They are able to select the most appropriate components that will meet their individual needs. Likewise, simulation users who use the component approach can select the most appropriate simulation components and models to meet the required training or analysis need. For example, a user could build a distributed simulation from ground combat, aerial, and naval sub-models to simulate large scale, joint operations. Similarly, the Army could build a high-resolution distributed simulation from infantry, tank, artillery, and helicopter sub-models to simulate ground combat. These two examples show the benefits of the component approach associated with distributed simulations by illustrating that the most appropriate components can be combined together to satisfy a particular need.

SIMNET had a very narrow focus that did not meet the needs of the various potential user groups. However, it provided the basic knowledge required to support future technological advances in model interoperability. It led to a series of distributed simulation support environments designed to support different distributed simulation user groups. Each group wanted fast, accurate, and resource conservative methods by which to communicate during distributed simulations. Each of the distributed simulation support environments defined its own set of standards. These separate user communities developed their own support environments tailored to meet only their specific needs. Simulation components that were capable of participating in a distributed simulation session in one environment were unavailable for use in another community's environment because of a lack of a universal standard. [22]

The two unique, protocol based distributed simulation support environments that emerged from SIMNET were Distributed Interactive Simulations (DIS) and Aggregate Level Simulation Protocol (ALSP). DIS was designed to support the constructive simulation community while ALSP was designed to support the aggregate simulation community. DIS generalized SIMNET technology by broadening its ability to represent military systems in synthetic environments. It met the military's need for capturing interactions between all types of combat entities and proved useful for analysis, training, mission rehearsal, and experimentation at the high-resolution level. It succeeded as a distributed simulation support environment because it operated a message-based environment with no central node, object ownership, offered a geographic distribution of components, and a common shared terrain representation. ALSP emerged after DIS and

served as a support environment capable of handling the Army's distributed simulations aggregated modeling needs. It extended the interoperability of DIS into the aggregate model community and enhanced detection, movement, and object interactions at this lower level of resolution. Like DIS, ALSP operated a message-based environment with no central node, object ownership, a geographic distribution of components, and a common terrain representation. It enhanced interoperability beyond the level available in DIS by providing better time, data, and architecture management functionality. [22]

The DoD modeling and simulation (M&S) community recognized that DIS, ALSP, and other more specialized simulation support environments failed to provide a general solution to the problem of interoperability. A general solution did not exist because each community tailored their particular distributed simulation support environment to meet community specific needs without regard for a general solution to meet the needs of other communities. Because these support environments duplicated modeling effort, lacked reusability, required overhead, and were wasteful, they were unable to provide a general solution to the issue of distributed simulation interoperability. Interoperability remained an issue because of the lack of one standard; further, it demonstrated the need for a general and standard distributed simulation support environment that used a common, accepted rule set. [22]

APPENDIX C. ARMY LEVELS OF ABSTRACTION

There are typically a number of layers of abstraction between the study question and the data required to answer that question. Figure 22 illustrates various levels of Army command and control (C2) components abstracted to apply to any study question. [10] It illustrates the levels of abstraction that must be addressed to decompose study questions into their lowest level of abstraction. The figure highlights changes to measurability through the levels of abstraction, and the roles of doctrine and analysts. Although the figure depicts an Army specific context, a similar schema is "generalizable" for any system or study question in any application area where the study question is far removed from the data.

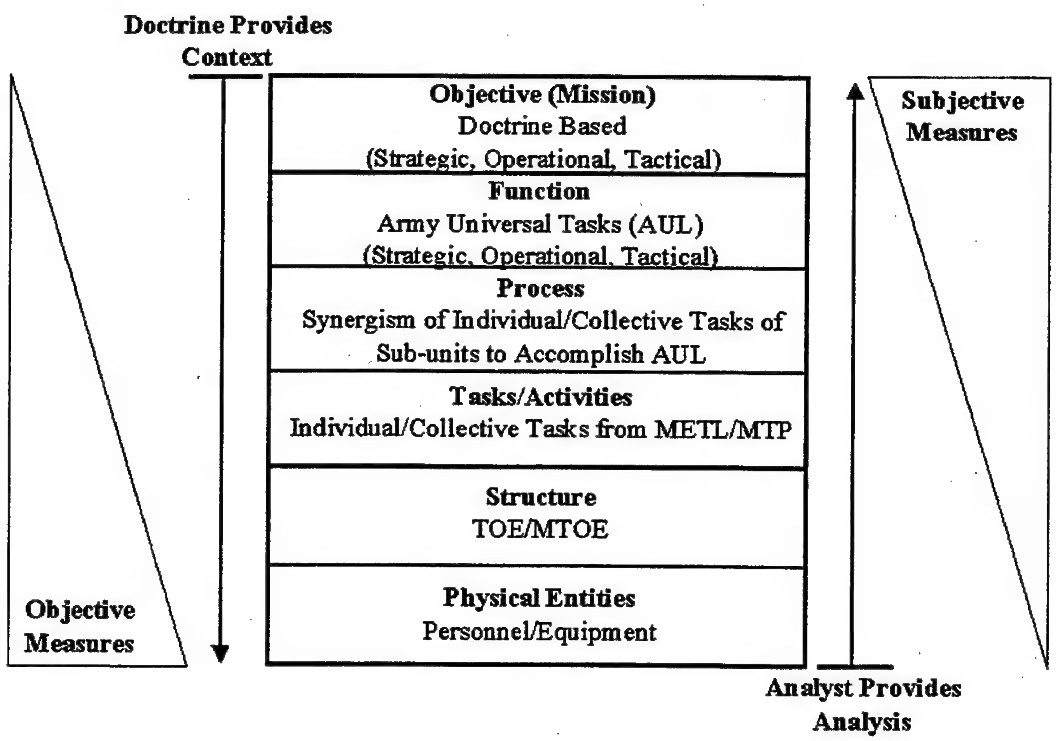


Figure 22. Army Levels of Abstraction

Figure 22's bottom tier, Physical Entities, provides the most objective measurability because it is generally easier to collect and analyze data on individual personnel and equipment performance than data on the entire Army. The Army uses the term Structure to describe the next level of abstraction. Examples of Structure include Tables of Organization and Equipment and Modified Tables of Organization and Equipment (TOE/MTOE) that describes the allocation of low level entities. The Army created these structures to equip its soldiers properly to perform both individual and collective tasks as part of an organization. The Army's next level of abstraction, Tasks/Activities, describes the actions of soldiers and their equipment in their war-fighting organizations. This level includes tasks necessary for mission accomplishment described by their Mission Essential Task Lists (METL) and Mission Training Plans (MTP). The Army uses the term Process to refer to the activities performed to fulfill the Functions articulated by doctrine. The Process level captures the synergies of individual and collective sub-units tasks. The next level of abstraction, Functions, aggregates all sub-levels of abstraction and describes what and how the Army achieves its objectives or doctrinally defined missions at the strategic, operational, and tactical levels of war. The highest level of abstraction, Objective (Mission), describes the overall end-state the Army desires in pursuing its objectives or missions. [10] Although this illustration describes an Army context, it is applicable to any organization that uses nested concepts to abstract its structure. The challenge for the analyst tasked to answer a study question is to identify at what level of abstraction the question is asked or resides and how best to decompose that

question into quantifiable data requirements. The SQM is a tool that can help provide the answer.

APPENDIX D. AH-64D FUTURE TECHNOLOGY DECOMPOSITION

This appendix describes the decomposition conducted for AH-64D firepower effectiveness in the future in more detail than Section III.C.1.c.

Having completed decomposition of the firepower implications with today's technology, the analyst must return to the branch of the tree that considers future technology (Figure 6). The analyst can now contemplate effectively employing firepower in an urban environment with future technology. The same quantification challenges exist along this branch as they did with today's technology.

While the analyst cannot know the impact of undiscovered techniques, using operational needs and context, he can articulate measures to assess its potential impact and compare these results to current technology. Simulations serve as a method by which to investigate these technologies, especially early in the development cycle, since they allow analysts to evaluate future technology.

The SQM provides the analyst structure from which to explore new technologies. Clearly, at this level of decomposition the analyst has not quantified or even identified particular technologies; however, the framework is dynamic enough to allow the analyst to add details at lower levels within the SQM tree. Again, the analyst may use current doctrine as the context from which to decompose into more quantifiable detail. Many of these processes remain unchanged even with conceived future technologies. Detection, classification and confirmation, BDA, and terminal effects all remain relevant with future technology. Specific procedures or parameters may change based on the technology itself.

The analyst in this illustration may first explore the effectiveness of direct fire weapons in urban environments. Further, he or she can focus attention on potential lethal and non-lethal direct fire weapons of the near future. The analyst may explore these types of weapons based on current interests of his or her decision-maker or organization. The analyst may investigate potential non-lethal, direct fire weapons that may prove useful in MOOTW or special operation scenarios. Using the SQM, he can decompose and develop MOEs/MOPs necessary to provide an assessment of the direct fire, non-lethal weapons as depicted in Figure 23.

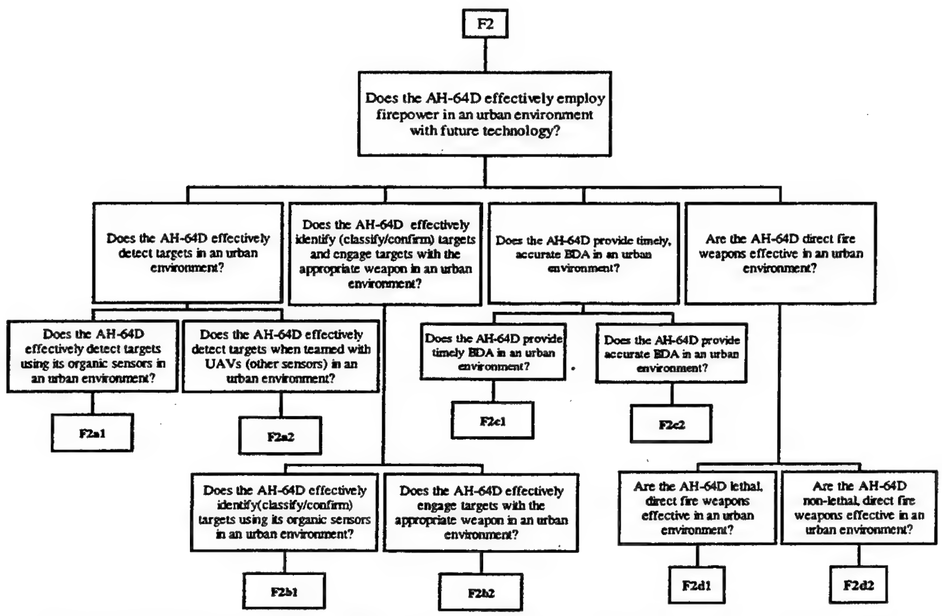


Figure 23. Employing Firepower in Urban Environments (Future)

The analyst may be interested in the ability to suppress, neutralize, or "destroy" targets, especially enemy air-defense assets, and minimizing collateral damage. These three terms serve as both battle tasks and target effects so the analyst must make a distinction when quantifying. This decomposition is similar to that developed under

“today’s lethal direct fires” (APPENDIX G) except that it slightly modifies the definitions of suppression, neutralization, and destruction effects for non-lethal targets.

Using a doctrinal context, the analyst may define suppression for an individual AH-64D firing at an enemy, non-lethal target. The analyst may define suppression as direct fire against enemy personnel, weapons, or equipment to prevent those targets from effectively firing on the AH-64D or other friendly forces. [23] The analyst may determine that the AH-64D successfully suppressed the target if the AH-64D maintains its own ability to maneuver (if suppression fired in self-defense). The analyst can add that the AH-64D allows the force for which it suppresses the target to extricate itself from the situation at hand. [19] Neutralization renders a target out of action by causing ten percent or greater casualties or damage to the target. [23] The analyst may use this standard to establish neutralization thresholds for non-lethal, direct fire weapons in urban environments. Finally, the analyst can establish “destruction” criteria. Since he or she addresses non-lethal weapons, destruction does not have the conventional meaning. Rather, in this context, destroy means "to physically render an enemy force or target so damaged that it cannot function as intended nor be restored to a usable condition without being entirely rebuilt.” [23] The analyst can quantify this by requiring thirty- percent casualties or material damage to the target.

The analyst may define effectiveness in terms of mission requirements. In this illustration (Figure 23), the analyst can identify the major difference between current and future technology effectiveness is in the type of weapon. The degree to which the non-lethal, direct fire weapon of interest meets or exceeds its “non-terminal” effects is still the

main interest whether talking about current or future technology. Does the weapon do what it proposes to do? The harder part of the process for the analyst may not be the application of the SQM but identifying existing tools to provide the data with which to answer the questions. The types of questions in this illustration remain similar.

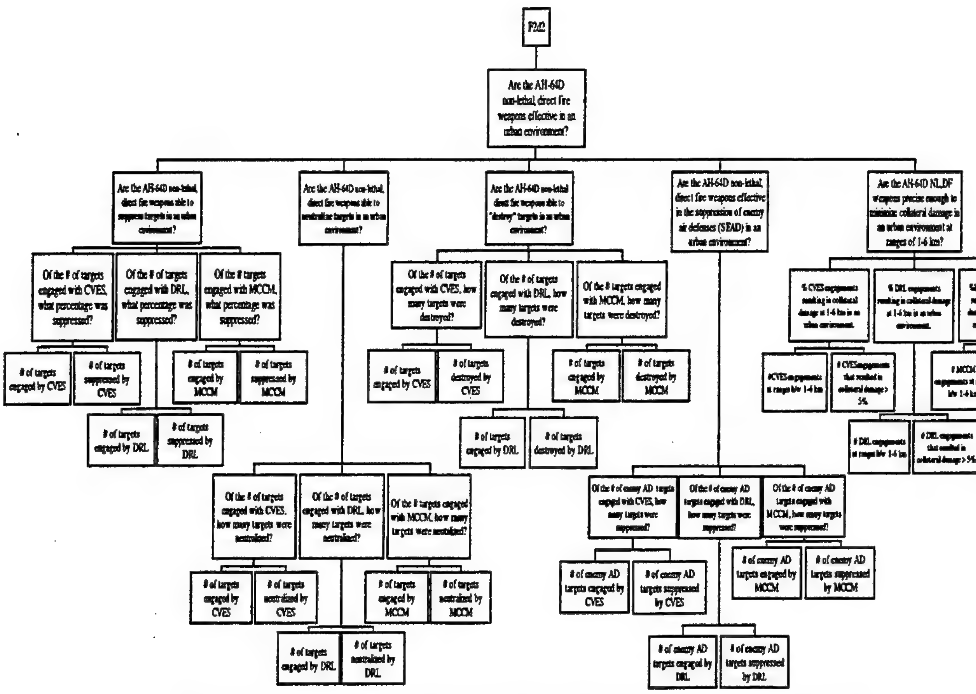


Figure 24. AH-64D Direct Fire Effectiveness (Future)

APPENDIX E. ASQM IN HLA

This appendix demonstrates how the "remote data filtering" concept of Chapter IV.B was abstracted to provide a general automation solution for the ASQM. Figure 25 depicts an HLA federation that consists of a publishing federate (FED1) and an analysis tool federate (FED2). The RTI executive interacts with the local RTIs of the two federates during the distributed simulation to provide distributed simulation services. Within FED2, four component processes (P1-P4) perform various functions. These components could exist as four separate processes or their functionality could potentially be consolidated in a variety of ways to reduce the number of separate processes. P1 represents the software tool used by the analyst to develop the SQM tree. This process allows the user to enter the three node characteristics for each node on the SQM tree. It includes provisions that automatically convert the equations and algorithms into logic that can be exported or executed remotely. P2 represents the automated HLA functionality that is provided by the Analysis Federate. P2 accomplishes this by communicating with the local RTI component. P1 communicates with P2 to ensure that the leaf node data subscription requirements are automatically implemented as part of the overall ASQM process. P3 represents the object-oriented/object hierarchy database process that stores the objects in their native FOM forms. It obtains the object hierarchy and the exercise data from P2. P4 is a separate analysis tool that has its own GUI that allows the user to easily generate arbitrary queries using a point and click interface. The ASQM capitalizes on this arbitrary query functionality. P4 receives logic from P1. This

logic includes the algorithms, the corresponding data queries, and the specified display format. The queries that are generated by P1 are in the format that is used by P4 to generate the arbitrary database queries. This consistency allows P4 to issue the queries to P3 that were generated by P1. The execution of this remote logic is automatic. The SQM is displayed and updated on a real-time basis by P4. P4 can still be used to accomplish its original design purposes while it is displaying and updating the SQM tree.

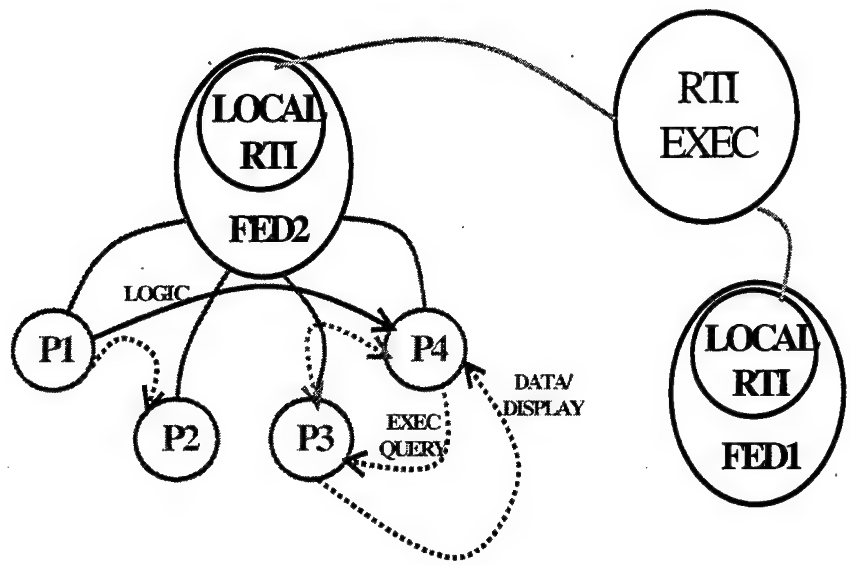


Figure 25. Proposal: Populating the ASQM Tree (Fully Automated)

APPENDIX F. DISCRETIZATION OF SOLUTION SET

The following section describes a method by which the ASQM could discretize the solution of any node set. The ASQM takes advantage of a concept known as "fuzzy set" theory that maps continuous data to discrete data. It "discretizes" answer sets into categories based on the question posed at each node during decomposition and identification of answer type. For example, each "question" or node of the SQM tree has an answer from a finite set of possible answers.

The proposed automated study question methodology addresses four parameter types: enumeration, Boolean, integer, and double.

Table 1. ASQM Common Data Types

Data Type	Data Values	Initial (Default)	Example Answer Set
Enumeration	Highly Relevant, Relevant, Not Relevant	Unknown	{Unknown, Highly Relevant, Relevant, Not Relevant}
Boolean	True, False	Unknown	{ Unknown, True, False}
Integer	..., -2, -1, 0, 1, 2, ...	0	{<0, 0..1, 2, 3..5, 6..10, >10}
Double	(-infinity, +infinity)	0.0	{(..0), [0], (0..1), [1,2), [2..4), [4..)}

Table 1 illustrates the four parameter types, their data values in a particular case, initial or default settings for those types, and examples of each. Enumeration valued questions (nodes) can exist in any state enumerated by the user and can include an unknown initial state. Boolean type questions (nodes) exist in states true or false but can also be unknown. Integer valued questions (nodes) partitioned into a finite number of

subsets can assume two or more states. Similarly, users can partition double valued questions (nodes) into a finite number of subsets as depicted. Using the ASQM, the analyst must enumerate combinations of child node answers needed to answer the parent question. In order to preclude computer I/O and other type errors, all questions (nodes) should have a default value corresponding to the initial state. Any node in the tree without an answer set will generate an error message unless addressed in the programming and enumeration. The user should establish this default setting through the ASQM GUI during decomposition, and confirm the partitioning during synthesis. [24]

The question, "Does the AH-64D effectively detect targets using its organic sensors in an urban environment?" was decomposed into five MOEs, each of which were further decomposed into numerous data level nodes. If the above question had only two child nodes with answers from real and Boolean values as described below, its enumerated answer set would be as depicted in Table 2.

Table 2. Enumeration Valued Mapping of Child Nodes

Enumeration Valued Mapping (Parent Node Answer Set)		Boolean Valued Question (Child Node)		
		True	False	Unknown
Double Valued Question (Child Node)	Unknown	Unknown	Unknown	Unknown
	[0.0]	Unknown	Not Effective	Unknown
	(0.0..0.4)	Not Effective	Not Effective	Unknown
	[0.4..0.6)	Effective	Not Effective	Unknown
	[0.6..0.75)	Effective	Effective	Unknown
	[0.75..0.9)	Effective	Effective	Unknown
	[0.9..1.0]	Highly Effective	Effective	Unknown

For example, in Table 2, the analyst maps (True, (0.0..0.4) and (False, [0.0..0.6)]) to "Not Effective." In this illustration, the user required certain conditions for a classification of not effective. Although the AH-64D detected up to 40% of the targets, and the crew detected the target within the prescribed time standard, the system was still not effective because it only detected up to 40%.

Table 3's domain includes six mappings that, together with the default settings of the child nodes, completely fill Table 2's 21 cells.

Table 3. Mapping Domain

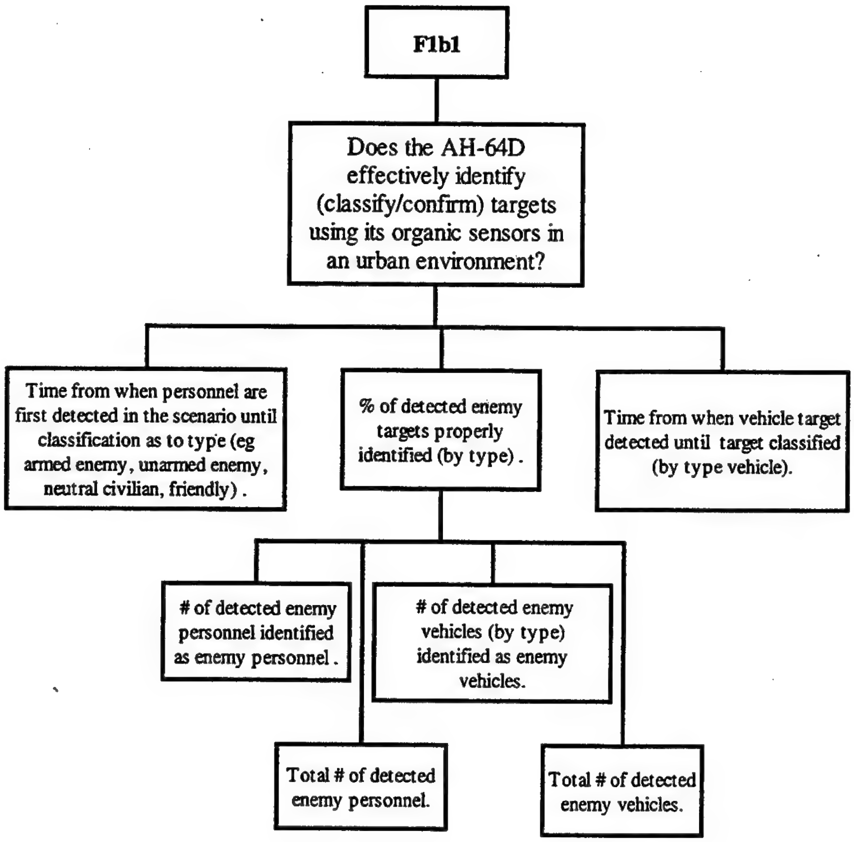
Maps To	Domain
Not Effective	(True, (0.0..0.4)
	(False, [0.0..0.6))
Effective	(True, [0.4..0.9)
	(False, [0.6..1.0])
Highly Effective	(True, [0.9..1.0])

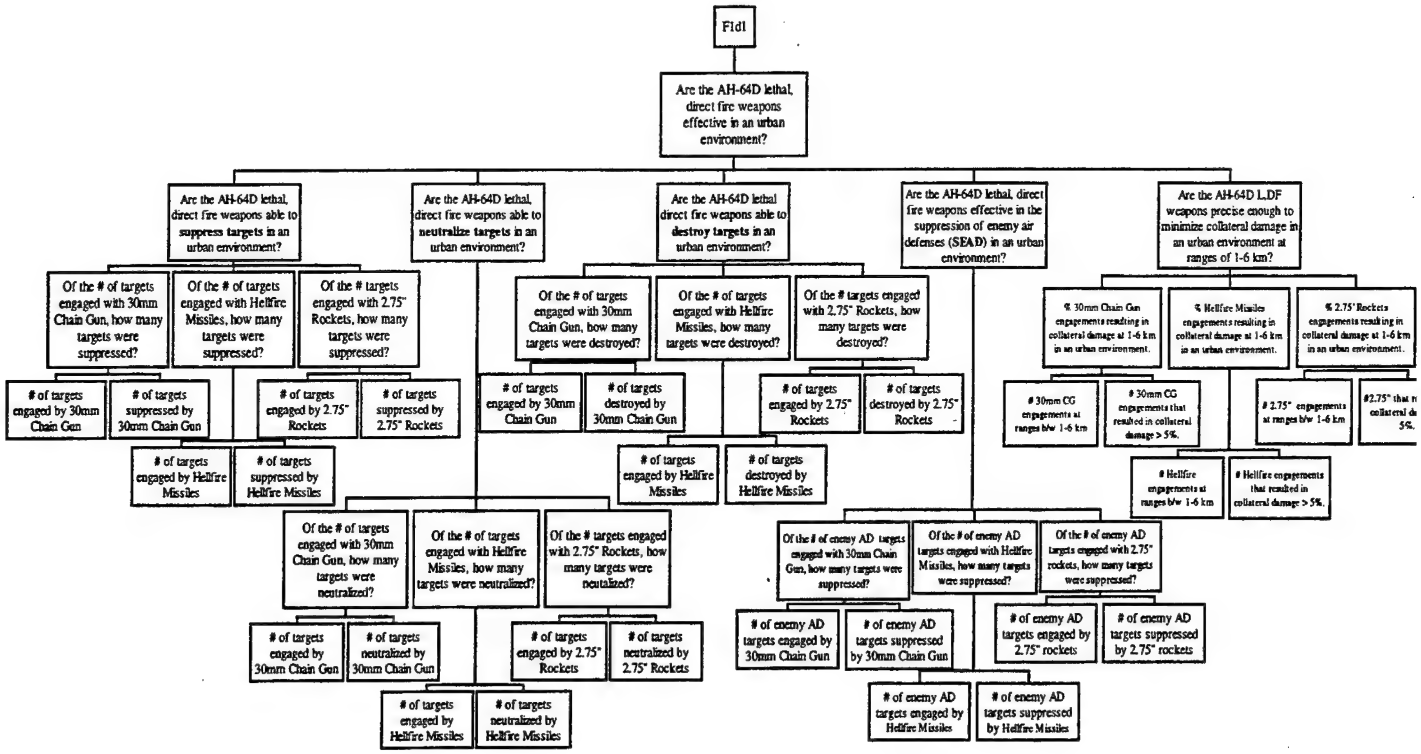
Since each question (node) has a finite set of possible answers, the programmer can store the answers as strings in a list form. By assigning natural numbers corresponding to the index of a particular answer in the list, the programmer can extract from the list by pointing to the natural number. The mapping functions for the data-level nodes (furthest depth of tree) assume the range of the data types into the natural numbers, indexes the answer with the result, and displays the answer. Other node mapping functions take an n-tuple of natural numbers into the natural numbers where there are n child nodes, indexes the answer with the result, and displays the result. [24]

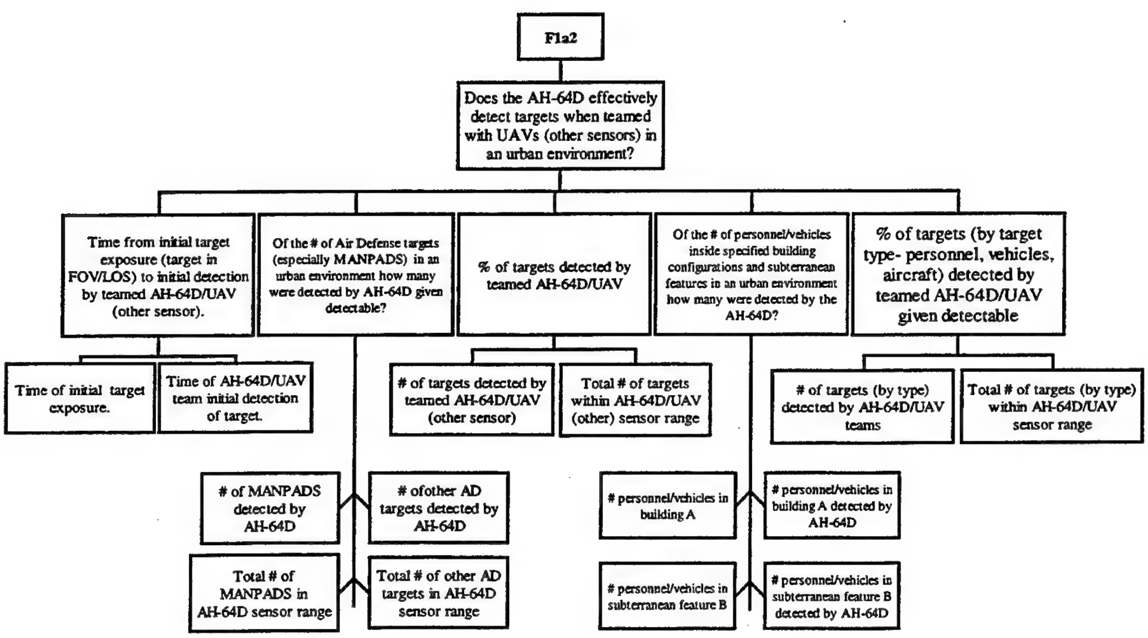
As discussed in Chapter IV, pivot tables may be a quick and easy way for the user of the ASQM to do this mapping. Spreadsheet like functionality could help users articulate the important relationships without burdening them with confusing interaction charts.

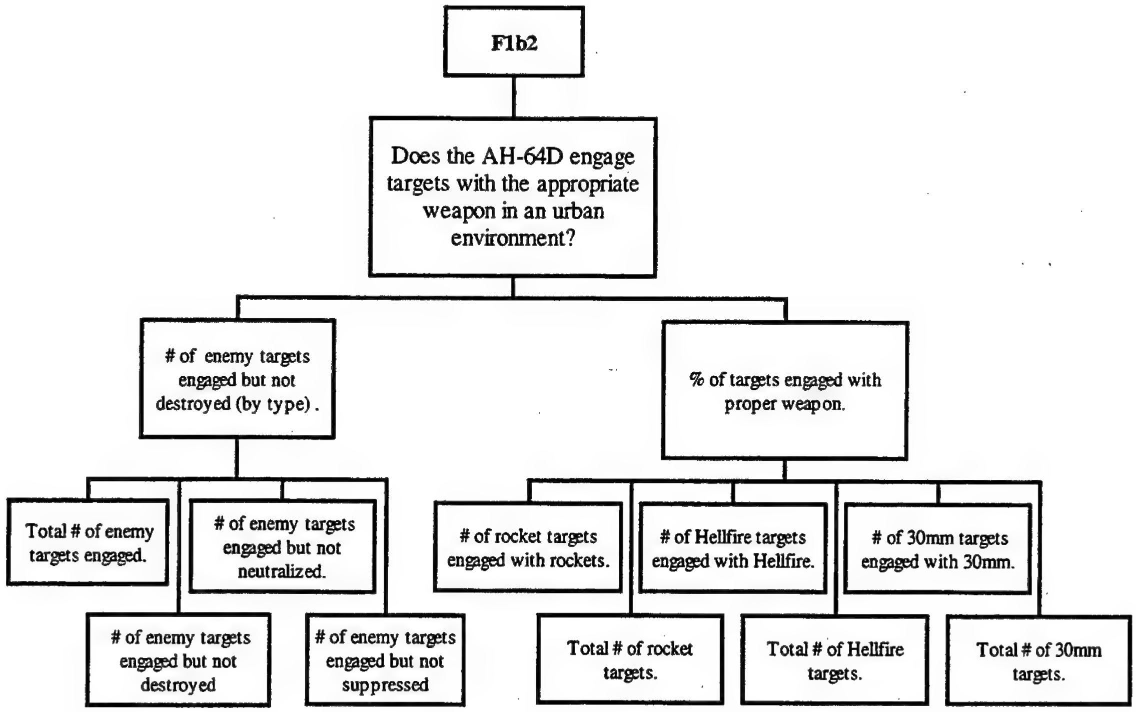
APPENDIX G. SQM TREES

This appendix includes samples from the SQM tree developed in the application chapter of this thesis. It does not include all branches of the SQM tree; rather, it only includes some branches of the tree.




F1d1
Are the AH-64D lethal, direct fire weapons effective in an urban environment?
Are the AH-64D lethal, direct fire weapons able to suppress targets in an urban environment?
Are the AH-64D lethal, direct fire weapons able to neutralize targets in an urban environment?
Are the AH-64D lethal, direct fire weapons able to destroy targets in an urban environment?
Are the AH-64D lethal, direct fire weapons effective in the suppression of enemy air defenses (SEAD) in an urban environment?
Are the AH-64D LDF weapons precise enough to minimize collateral damage in an urban environment at ranges of 1-6 km?
Of the # of targets engaged with 30mm Chain Gun, how many targets were suppressed?
Of the # of targets engaged with Hellfire Missiles, how many targets were suppressed?
Of the # of targets engaged with 2.75" Rockets, how many targets were suppressed?
# of targets engaged by 30mm Chain Gun
# of targets suppressed by 30mm Chain Gun
# of targets engaged by 2.75" Rockets
# of targets suppressed by 2.75" Rockets
# of targets engaged by Hellfire Missiles
# of targets suppressed by Hellfire Missiles
Of the # of targets engaged with 30mm Chain Gun, how many targets were destroyed?
Of the # of targets engaged with Hellfire Missiles, how many targets were destroyed?
Of the # targets engaged with 2.75" Rockets, how many targets were destroyed?
# of targets engaged by 30mm Chain Gun
# of targets destroyed by 30mm Chain Gun
# of targets engaged by 2.75" Rockets
# of targets destroyed by 2.75" Rockets
# of targets engaged by Hellfire Missiles
# of targets destroyed by Hellfire Missiles
% 30mm Chain Gun engagements resulting in collateral damage at 1-6 km in an urban environment.
% Hellfire Missiles engagements resulting in collateral damage at 1-6 km in an urban environment.
% 2.75" Rockets engagements resulting in collateral damage at 1-6 km in an urban environment.
# 30mm CG engagements at ranges b/w 1-6 km
# 30mm CG engagements that resulted in collateral damage > 5%.
# 2.75" engagements at ranges b/w 1-6 km
# Hellfire engagements at ranges b/w 1-6 km
# Hellfire engagements that resulted in collateral damage > 5%.
Of the # of targets engaged with 30mm Chain Gun, how many targets were neutralized?
Of the # of targets engaged with Hellfire Missiles, how many targets were neutralized?
Of the # targets engaged with 2.75" Rockets, how many targets were neutralized?
# of targets engaged by 30mm Chain Gun
# of targets neutralized by 30mm Chain Gun
# of targets engaged by 2.75" Rockets
# of targets neutralized by 2.75" Rockets
# of targets engaged by Hellfire Missiles
# of targets neutralized by Hellfire Missiles
Of the # of enemy AD targets engaged with 30mm Chain Gun, how many targets were suppressed?
Of the # of enemy AD targets engaged with Hellfire Missiles, how many targets were suppressed?
Of the # of enemy AD targets engaged with 2.75" rockets, how many targets were suppressed?
# of enemy AD targets engaged by 30mm Chain Gun
# of enemy AD targets suppressed by 30mm Chain Gun
# of enemy AD targets engaged by 2.75" rockets
# of enemy AD targets suppressed by 2.75" rockets
# of enemy AD targets engaged by Hellfire Missiles
# of enemy AD targets suppressed by Hellfire Missiles


F1a2
Does the AH-64D effectively detect targets when teamed with UAVs (other sensors) in an urban environment?
Time from initial target exposure (target in FOV/LOS) to initial detection by teamed AH-64D/UAV (other sensor).
Of the # of Air Defense targets (especially MANPADS) in an urban environment how many were detected by AH-64D given detectable?
% of targets detected by teamed AH-64D/UAV
Of the # of personnel/vehicles inside specified building configurations and subterranean features in an urban environment how many were detected by the AH-64D?
% of targets (by target type- personnel, vehicles, aircraft) detected by teamed AH-64D/UAV given detectable
Time of initial target exposure.
Time of AH-64D/UAV team initial detection of target.
# of targets detected by teamed AH-64D/UAV (other sensor)
Total # of targets within AH-64D/UAV (other) sensor range
# of targets (by type) detected by AH-64D/UAV teams
Total # of targets (by type) within AH-64D/UAV sensor range
# of MANPADS detected by AH-64D
ofother AD targets detected by AH-64D
Total # of MANPADS in AH-64D sensor range
Total # of other AD targets in AH-64D sensor range
personnel/vehicles in building A
personnel/vehicles in building A detected by AH-64D
personnel/vehicles in subterranean feature B
personnel/vehicles in subterranean feature B detected by AH-64D


F1b2
Does the AH-64D engage targets with the appropriate weapon in an urban environment?
# of enemy targets engaged but not destroyed (by type).
% of targets engaged with proper weapon.
Total # of enemy targets engaged.
# of enemy targets engaged but not neutralized.
# of enemy targets engaged but not destroyed
# of enemy targets engaged but not suppressed
# of rocket targets engaged with rockets.
# of Hellfire targets engaged with Hellfire.
# of 30mm targets engaged with 30mm.
Total # of rocket targets.
Total # of Hellfire targets.
Total # of 30mm targets.

APPENDIX H. MOE/MOP CATALOG

This Appendix lists some examples of measures of effectiveness or performance identified in the application chapter of this thesis. The intent is to provide more detail into the MOE/MOP used. These MOE/MOPs are not exhaustive and require significant additional work in order to prove more useful to analyses, tests, evaluations, and research. As discussed in the body of this thesis, the analyst is responsible for developing appropriate MOE/MOPs for his or her particular study question. This format may prove useful should the ASQM allow for a library of trees from which analyst could extract and modify relevant branches. The following format proved useful in the development of the SQM and its application. The author extracted the following format from a TRADOC Technical Document. [10]

Definition of the measure	A complete statement of the measure that includes computational data and methods of processing.
Dimensions of the measure	How the measure is expressed (level and unit of measure).
Levels of measure	Include nominal, ordinal, interval, and ratio. Examples of units of measure include integer, real; green, amber, red; high, medium, low; kilometers per hour; 80 percent.
Limits on the range of measure	Statement of any limits on input or output of the measure.
Rationale for the measure	Why the measure was selected and what properties make it useful.
Relevance of the measure	Circumstances (analyses, studies, etc) in which the measure would contribute to the decision process.
Associated measures	Other measures that either may be used in conjunction with the measure or that must be used with it to evaluate appropriately the issue.
Applications	Studies, tests, or evaluations in which the measure was used or observed.
References	Sources that provide any additional discussion of the measure and its use.

Context: Does the AH-64D effectively employ firepower in an urban environment with today's technology?

Sub-Issue: Does the AH-64D effectively detect targets in an urban environment?

EEA: Does the AH-64D effectively detect targets using its organic sensors in an urban environment?

Definition of the measure. Time to detect targets (DetTime) from when the target is initially detectable by the AH-64D until the AH-64D actually detects the target. The target is considered detectable if line of sight (LOS) exists between the Apache and the target. Input data are the initial target exposure time (intTgtExpTime) and the initial AH-64D detection time (intTgtDetTime). Relation of output to input is:

DetTime = intTgtDetTime – intTgtExpTime

Dimensions of the measure. Time – output is a double expressing the time it takes the Apache to detect targets in an urban environment.

Limits on the range of measure. The output may vary from zero to infinity. It could only be zero if the Apache immediately detected the target as it became detectable. The output could only be infinity if the target was not detected by the Apache.

Rationale for the measure. This measure addresses the first step of the target engagement process. Targets must be detected before action is taken against the target. It is assumed that more effective sensor systems (including the crew, organic sensors like the TADS, etc) require less time to detect targets. The output will be compared to standards from Apache Gunnery Tables to ultimately return a value to the parent EEA.

Relevance of the measure. The measure is used to evaluate detection effectiveness.

Associated measures. % of detentions within certain thresholds

Applications. used in Apache SQM demonstration.

References. FM 1-140 Helicopter Gunnery, 29 March 1996

Context: Does the AH-64D effectively employ firepower in an urban environment with today's technology?

Sub-Issue: Does the AH-64D effectively detect targets in an urban environment?

EEA: Does the AH-64D effectively detect targets using its organic sensors in an urban environment?

Definition of the measure. Percentage of targets (by type target) detected by AH-64D given that the target was detectable ($PerTgtDet_t$, t = personnel, vehicles, and aircraft). Targets are considered detectable if LOS is achieved. Inputs include the number of targets by type ($numTgtDet_t$, t= personnel, vehicles, and aircraft) that the AH-64D detected and the total number of targets ($numTgtDet_{TOT}$, TOT= all targets in LOS with AH-64D over the specified time period). Relation of output to input is:

$$PerTgtDet_t = (numTgtDet_t / numTgtDet_{TOT}) * 100$$

Dimensions of the measure. Ratio – output is a percentage in terms of percentage of detections by type target.

Limits on the range of measure. The output can assume any value from zero to one hundred percent.

Rationale for the measure. This measure addresses quality of the sensor systems indirectly by assessing the effectiveness of detection of particular target types. It is assumed a more effective sensor system will produce a higher percentage of detentions. The output will be compared to standards extrapolated from Apache Gunnery Tables to ultimately return a relative scaling of effectiveness to the parent EEA.

Relevance of the measure. The measure is used to evaluate detection effectiveness.

Associated measures. Time to detection….

Applications. used in Apache SQM demonstration.

References. FM 1-140 Helicopter Gunnery, 29 March 1996

EEA: Does the AH-64D effectively detect targets using its organic sensors in an urban environment?

Definition of the measure. Percentage of targets detected by AH-64D by target type ($PerTgtDetAH64D_t$, t = personnel, vehicles, and aircraft) versus other platform types ($PerTgtDetOther_t$, t = personnel, vehicles, and aircraft). The measure returns a percentage. Targets are considered detectable if LOS is achieved. Inputs include the number of targets by type ($numTgtDet_t$, t= personnel, vehicles, and aircraft) that the AH-64D detected, the number of targets by type detected by all other sensors ($numTgtDetOther_t$, t = personnel, vehicles, and aircraft; Other = UAV, ground forces, etc.), the total number of targets ($numTgtDet_{TOT}$, TOT= all targets in LOS with AH-64D; $numTgtDetOther_{TOT}$ = all targets in LOS with other sensor types over the specified time period). Relation of output to input is:

$PerTgtDet_t = (numTgtDet_t / numTgtDet_{TOT}) * 100$

$PerTgtDetOther_t = (numTgtDetOther_t / numTgtDetOther_{TOT}) * 100$

$PerTgtDetAH64D_t = PerTgtDet_t / PerTgtDetOther_t$

Dimensions of the measure. Ratio – output is a percentage in terms of percentage of detections by type target.

Limits on the range of measure. The output can assume any value from zero to one hundred percent. The measure is not very refined in that it ignores the availability of target types to different sensors. For example, if the AH-64D achieved a 90% detection rate but only had 10 detectable targets, and "Other" sensors achieved a 50% detection rate but had 100 detectable targets, the observer may believe the AH-64D is more effective in detecting targets in an urban environment without considering the opportunities to sense targets. A more refined measure could be constructed to take other issues into account.

Rationale for the measure. This measure indirectly addresses the role of the AH-64D in detecting targets in an urban environment versus other sensor types. It is assumed a more effective sensor system will produce a higher percentage of detentions. In comparing the AH-64D to other methods of detection, it is hoped that the analyst will gain insight into what sensors are used effectively in detecting various. target types in an urban environment. The output will be compared but given little weight in the parent EEA due to the variability of the issue. Its purpose is more illustrative than deterministic.

Relevance of the measure. The measure evaluates detection effectiveness.

Associated measures. Time to detection….

Applications. used in Apache SQM demonstration.

References. None

APPENDIX I. AH-64D LONGBOW APACHE REFERENCES

Longbow Specifications:

Copyright © 1999 The Boeing Company - All rights reserved
http://www.boeing.com/rotorcraft/military/ah64d/ah64lbtech.htm

Copyright © 1999 The Boeing Company - All rights reserved
http://www.boeing.com/rotorcraft/military/ah64d/ah64dtech.htm

LIST OF REFERENCES

1. Defense Modeling and Simulation Office (DMSO), "HLA," [http://hla.dmso.mil/hla/main.htm].

2. Defense Modeling and Simulation Office (DMSO), "What is HLA," [http://hla.dmso.mil/].

3. Defense Modeling and Simulation Office (DMSO), "DODD 5000.59, DoD Modeling and Simulation (M&S) Management." [http://web7.whs.osd.mil/text/d500059p.txt]. 4 January 1994.

4. Knight, Steven D., *A Comparison of Analysis in DIS and HLA*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1998.

5. Pearman, Gerald M., *Comparison Study of Janus and Jlink*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1997.

6. Jackson, Leroy A. MAJ, and Wood, J. Ralph LTC, *Exploiting the High Level Architecture for Analysis in Advanced Distributed Simulation*, TRADOC Analysis Center-Monterey, 1997.

7. Defense Modeling and Simulation Office, "HLA Rules, Version 1.0", 21 August 1996.

8. Murphy, Jr. William S. MAJ, *Analysis Federate*, TRADOC Analysis Center-Monterey, 1997.

9. Training and Doctrine Command (TRADOC) Pamphlet No 11-8 (Draft 1995), *Army Programs, TRADOC Studies and Analyses Pamphlet*, 1995.

10. Training and Doctrine Command (TRADOC), *Command and Control Measures of Effectiveness Handbook (C2MOE Handbook)*, Technical Document TRAC-TD-0393, October 1993.

11. The Army Homepage defining Army Missions, [http://www.army.mil/mission_vision.htm].

12. Training and Doctrine Command (TRADOC), "Requirements Determination," General Dennis J. Reimer. [http://192.111.52.19/cmdpubs/reqdef.htm].

13. Marine Corps Urban Warrior, "Why Urban Warrior." [http://www.mcwl.org/mcwl/uw/why/why.htm].

14. Army Field Manual No. (FM) 90-10-1, *An Infantryman's Guide to Combat In Built-Up Areas*, Headquarters, Department of the Army, 12 May 1993.

15. Parry, Samuel, OA4655, Joint Combat Modeling [Class notes and personal discussions], Naval Postgraduate School, Monterey, California, Fall 1998.

16. Army Field Manual No. (FM) 1-112, *Attack Helicopter Operations*, Headquarters, Department of the Army, 2 April 1997.

17. The Website for Defense Industries, "Army, Current Projects." [http://www.army-technology.com/projects/apache/].

18. Department of the Army, Pamphlet 11-XX (Draft), *Army Universal Task List*, 1990.

19. Army Field Manual No. (FM) 1-140, *Helicopter Gunnery,* Headquarters, Department of the Army, 2 April 1997.

20. Murphy, Jr. William S. MAJ, TRADOC Analysis Center, and Aswegan, Galen D. Tapestry Solutions, INC., *High Level Architecture Remote Data Filtering*, Winter Simulation Conference, 1998.

21. Kuhn, T.S., *The Structure of Scientific Revolutions*, 3d ed., The University of Chicago Press: Chicago, Illinois, 1996.

22. Murphy, Jr. William S. MAJ, *A Conceptual Framework for Model Composability in Architecture Based Distributed Simulation Support Environments*, Dissertation Proposal, February 1999.

23. Army Field Manual No. (FM) 101-5-1, *Operational Terms and Graphics*, Headquarters, Department of the Army, 30 September 1997.

24. Jackson, Leroy A. MAJ, and Rauhut, Michael W. CPT, *Automating the Study Question Methodology*, [Working paper], TRADOC Analysis Center-Monterey, 1999.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Fort Belvoir, Virginia 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Professor Arnold H. Buss, Code OR/Bu 2
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943-5000

4. Dr. Samuel Parry, Code OR 1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943-5000

5. Major William S. Murphy Jr. 2
   PO Box 8692
   Monterey, California 93943-0692

6. TRADOC Analysis Center 1
   ATTN: ATRC-TD
   Fort Leavenworth, Kansas 66027-2345

7. TRADOC Analysis Center Research Activities 1
   PO Box 8692
   Monterey, California 93943-0692

8. CPT Michael W. Rauhut 3
   3551 Willow Road
   Cuba, New York 14727